

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"  
УДК 004.78

«До захисту допущено»  
Завідувач кафедри  
О.В. Коваль  
(підпис) (ініціали, прізвище)  
“ ” 2019р.

**Магістерська дисертація**

зі спеціальності – 122 Комп'ютерні науки  
за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання  
процесів і систем  
на тему «Програмна реалізація людино-машинного інтерфейсу для візуалізації  
KNX інсталяцій»

---

Виконав: студент 6-го курсу, групи ТМ-81мп

Яшин Андрій Сергійович  
(прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Керівник доцент, к.е.н Сегеда І.В.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Рецензент \_\_\_\_\_  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності – 122 Комп’ютерні науки

за спеціалізацією - Комп’ютерний моніторинг та геометричне моделювання процесів і систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ О.В. Коваль

(підпис)

” \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ  
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Яшину Андрію Сергійовичу

(прізвище, ім’я, по батькові)

1. Тема дисертації Програмна реалізація людино-машинного інтерфейсу для візуалізації KNX інсталяцій

Науковий керівник Сегеда Ірина Василівна к.е.н, доцент

(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджені наказом по університету від 4 листопада 2019 року №3812-с

2. Строк подання студентом роботи \_\_\_\_\_

3. Об’єкт дослідження структура KNX-мережі

4. Предмет дослідження інструменти для візуалізації структури KNX-мережі, контролю та моніторингу її компонентів

5. Перелік питань, які потрібно розробити \_\_\_\_\_

1) проаналізувати структуру KNX-мережі та засоби її проектування;

2) проаналізувати проблематику проектування інтерфейсу користувача та принципи проектування інтерфейсу користувача;

3) сформулювати та проаналізувати базову функціональність для візуалізації структури KNX-мережі, контролю та моніторингу її компонентів;

4) розробити програмне забезпечення для розробки людино-машинного інтерфейсу для візуалізації структури KNX-мережі, використовуючи засоби Node-RED

6. Орієнтований перелік ілюстративного матеріалу \_\_\_\_\_

1) Функціональність програмного забезпечення; \_\_\_\_\_

2) Взаємодія компонентів системи; \_\_\_\_\_

3) Структура програмного забезпечення; \_\_\_\_\_

4) Інтерфейс технічного спеціаліста. \_\_\_\_\_

5) Інтерфейс кінцевого користувача. \_\_\_\_\_

7. Орієнтований перелік публікацій \_\_\_\_\_

Сегеда І. В., Яшин А.С., “Візуалізації структури KNX-мережі з використанням  
людино-машинного інтерфейсу”

8. Дата видачі завдання « 29 » вересня 2018 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Отримання завдання	29.09.2018	
2.	Збір інформації	09.10.2018 – 25.01.2019	
3.	Аналіз вимог завдання, вибір методів і засобів розв’язання поставленої задачі	26.01.2019 – 05.09.2019	
4.	Підготовка публікацій	19.07.2019, 02.09.2019	
5.	Підготовка доповідей на конференції	15.08.2019, 20.10.2019	
6.	Підготовка дисертації	03.07.2019 – 07.12.2019	
7.	Розробка програмного продукту	10.01.2019 – 09.10.2019	
8.	Захист програмного продукту	24.10.2019	
9.	Передзахист	28.11.2019	
10.	Захист	18.12.2019	

Студент \_\_\_\_\_ Яшин А.С.  
(підпис) (прізвище та ініціали,)

Керівник роботи \_\_\_\_\_ Сегеда І.В.  
(підпис) (прізвище та ініціали,)

# РЕФЕРАТ

Магістерська дисертація складається зі вступу, 6 основних розділів, висновку, переліку посилань з 25 найменувань, 1 додатка, 30 рисунків та 22 таблиць. Повний обсяг магістерської дисертації складає 78 сторінок, з яких перелік посилань займає 3 сторінки, а додатки – 4 сторінки.

У сучасному світі кожен очікує більшого комфорту та зручності, як вдома, так і на роботі. В даний час KNX - одне з найпоширеніших рішень для використання в середніх і великих системах автоматизації в будинках, офісах та комерційних приміщеннях. Вимоги до комфорту, безпеки та гнучкості сучасних будівель постійно зростають. У той же час, враховуючи постійний технологічний розвиток та посилення автоматизації систем комфорту, найважливішим питанням є економія електроенергії. Ось чому попит на інтелектуальні системи, що відповідають цим умовам, постійно зростає.

Використання технології KNX може надати рішення, які при використанні звичайних методів створення подібних систем можуть бути реалізовані лише насилу. Усі додатки в квартирі або будинку можна контролювати за допомогою однієї сенсорної панелі. Починаючи від систем опалення, вентиляції та контролю доступу, а також дистанційного керування всіма побутовими приладами - KNX пропонує абсолютно нові способи підвищення комфорту, безпеки та економії енергії в квартирах та будинках.

Метою дипломної роботи є розробка програмної реалізації людино-машинного інтерфейсу для візуалізації структури KNX-мережі та можливості швидкого налаштування алгоритму взаємодії компонентів мережі.

Для візуалізації структури програмного забезпечення різних мереж KNX, керування їх компонентами та моніторингу даних для подальшого аналізу та визначення більш оптимальних способів використання мережі, програмне забезпечення повинно забезпечувати такі функції:

- встановлення зв'язку між Node-RED та KNX мережею;
- відображення елементів управління мережі;

- управління та налаштування алгоритму взаємодії компонентів KNX мережі через інтерфейс Node-RED;
- управління станом модулів мережі через веб-інтерфейс кінцевого користувача;
- підключення до мережі;
- відображення історії користування;
- відображення подій мережі у реальному часі.

Для досягнення поставленої задачі були сформульовані наступні завдання дослідження:

- провести аналіз проблеми;
- порівняти існуючі рішення;
- виділити основні особливості таких систем;
- спроектувати архітектуру програмного забезпечення;
- розробити програмне забезпечення.

Об'єктом дослідження є структура KNX-мережі.

Предметом дослідження є інструменти для візуалізації структури KNX-мережі та швидкого налаштування алгоритму взаємодії компонентів мережі.

Практичним значенням одержаних результатів є можливість подальшої експлуатації розробленого програмного продукту на підприємствах чи у компанія.

Ключові слова: KNX, NODE-RED, ІНТЕРФЕЙС КОРИСТУВАЧА, МОНІТОРИНГ, МЕРЕЖА, МАСШТАБУВАННЯ.

## ABSTRACT

The master's dissertation consists of an introduction, 6 main sections, conclusion, a list of references of 25 names, 1 appendici, 30 figures and 22 tables. The full volume of the master's thesis is 78 pages, of which the list of links occupies 3 pages, and the annexes - 4 pages.

In today's world, everyone expects more comfort and convenience, both at home and at work. Currently KNX is one of the most common solutions for use in medium to large automation systems in homes, offices and commercial premises. Requirements for the comfort, safety and flexibility of modern buildings are constantly increasing. At the same time, given the ongoing technological development and increasing automation of comfort systems, the most important issue is energy savings. That is why the demand for intelligent systems that meet these conditions is constantly increasing.

The use of KNX technology can provide solutions that, when using conventional methods of creating such systems, can only be implemented with difficulty. All applications in an apartment or home can be controlled by a single touchpad. From heating, ventilation and access control, as well as remote control of all household appliances - KNX offers completely new ways to increase comfort, safety and energy savings in apartments and homes

The aim of the master's dissertation is to develop a software implementation of human-machine interface to visualize the structure of KNX-network and the ability to quickly adjust the algorithm for the interaction of network components.

To visualize the software structure of different KNX networks, manage their components and monitor data for further analysis and identify better ways to use the network, the software should provide the following features:

- linking Node-RED to KNX network;
- display of network controls;
- management and adjustment of the algorithm of interaction of KNX network components through the Node-RED interface;

- managing the status of the network modules through the end-user web interface;
- connection to the network;
- display of usage history;
- real-time network events display.

To achieve this task, the following research objectives were formulated:

- to analyze the problem;
- compare existing solutions;
- highlight the main features of such systems;
- design the software architecture;
- develop software.

The object of the study is the structure of the KNX network.

The subject of the study is tools to visualize the structure of the KNX network and to quickly configure the algorithm of interaction of network components.

The practical value of the obtained results is the possibility of further exploitation of the developed software product at the enterprises or the company.

Keywords: KNX, NODE-RED, USER INTERFACE, MONITORING, NETWORK, ZOOMING.

# ЗМІСТ

Перелік термінів, умовних позначень та символів .....	10
Вступ .....	11
1. Постановка задачі магістерської дисертації .....	13
Висновки до розділу 1 .....	14
2. Способи реалізації візуалізації KNX мереж .....	15
2.1. Користувацький інтерфейс .....	15
2.2. Проблеми проектування інтерфейсів користувача .....	17
2.3. Принципи проектування інтерфейсів користувача .....	18
2.4. Комунікаційна шина KNX .....	20
2.5. Веб-система Node-RED .....	27
Висновки до розділу 2 .....	28
3. Загальний опис інструментів реалізації програмного забезпечення .....	30
3.1. Середовище розробки WebStorm JetBrains .....	30
3.2. Мова програмування JavaScript .....	32
3.3. Платформа Node.js .....	34
3.4. Веб-платформа Node-RED .....	36
3.5. Бібліотека-розширення jQuery .....	39
3.6. Розмітка веб-сторінки HTML .....	40
3.7. Таблиці стилів CSS .....	41
Висновки до розділу 3 .....	43
4. Опис програмної реалізації .....	44
4.1. Адаптер між Node-RED та KNX мережею .....	44
4.2. Веб-інтерфейс кінцевого користувача .....	47
Висновки до розділу 4 .....	48
5. Методи взаємодії з веб-системою .....	49



5.1. Системні вимоги .....	49
5.2. Інсталяція системи.....	49
5.3. Сценарій роботи користувача з системою .....	50
Висновки до розділу 5 .....	56
6. Розробка стартап проекту.....	57
6.1. Опис ідеї стартап проекту.....	57
6.2. Технологічний аудит ідеї проекту .....	59
6.3. Аналіз ринкових можливостей запуску стартап проекту.....	59
6.4. Розробка ринкової стратегії проекту .....	66
Висновки до розділу 6 .....	71
Висновки .....	72
Список використаних джерел .....	74
Додаток А.....	77

## ПЕРЕЛІК ТЕРМІНІВ, УМОВНИХ ПОЗНАЧЕНЬ ТА СИМВОЛІВ

<b>JS</b>	JavaScript
<b>CSS</b>	Каскадні таблиці стилів
<b>HTML</b>	Мова розмітки гіпертекстових документів
<b>NODE.JS</b>	Програмна платформа для мови програмування JavaScript
<b>NODE-RED</b>	Веб-система для візуального програмування
<b>KNX</b>	Комунікаційна шина
<b>JQUERY</b>	Бібліотека-розширення для мови програмування JavaScript
<b>IK</b>	Інтерфейс користувача
<b>NPM</b>	Менеджер додатків для JavaScript
<b>LINUX</b>	UNIX-подібна операційна система

## ВСТУП

У сучасному світі кожен очікує більше комфорту та зручності, як вдома, так і на роботі. На сьогодні KNX - одне з найпоширеніших рішень для використання в середніх і великих системах автоматизації для будинків, офісів та комерційних приміщень. Також зростають вимоги до зручності, безпеки та гнучкості сучасних будівель. У той же час, враховуючи постійний технологічний розвиток та посилення автоматизації систем комфорту, найважливішим питанням є економія електроенергії. Ось чому попит на інтелектуальні системи, які відповідають цим умовам, постійно зростає.

За допомогою технологій KNX можна надати рішення, які важко реалізувати за допомогою звичайних методів створення таких систем. Усі додатки в квартирі чи будинку можна контролювати за допомогою сенсорної панелі. Від систем опалення, вентиляції та контролю доступу до дистанційного керування всіма побутовими електроприладами, KNX відкриває абсолютно нові можливості для підвищення комфорту, безпеки та енергоефективності в будинках та будинках.

Для візуалізації мережі KNX, управління і моніторингу її вузлів був розроблений людино-машинний інтерфейс та описаний алгоритм взаємодії компонентів мережі за допомогою інструментів Node-RED системи. Це дозволяє візуалізувати необмежену кількість мереж, управляти їх елементами і відслідковувати події в мережі. Це дозволяє досліджувати і поліпшити способи використання мереж KNX. Веб-додаток призначений як для техніків мережі KNX, так і для некваліфікованих користувачів.

Для розробки інтерфейсу веб-додатку було використано JavaScript та бібліотеку jQuery. Для реалізації частини адаптера програмного продукту використано Node.js та веб-систему графічного програмування Node-RED. Середовищем реалізації адаптера між KNX мережею та Node-RED системою було обрано WebStorm від компанії JetBrains.

У першому розділі “Постановка завдання магістерської роботи” сформульовано

мету й описано постановку задачі. Описуються основна мета роботи, вхідні та вихідні дані, а також методики розв’язання даної проблеми.

Другий розділ “Способи реалізації візуалізації KNX мереж” дає опис структури KNX мереж та надає загальний опис веб-системи для графічного програмування Node-RED.

У третьому розділі “Опис основних інструментів програмної реалізації” аналізуються інструменти, за допомогою яких можна вирішити поставлену задачу та реалізувати веб-систему для взаємодії з KNX мережею.

Четвертий розділ “Опис програмної реалізації” надає опис архітектури програмного забезпечення та алгоритму взаємодії компонентів системи.

У п’ятому розділі “Методика роботи користувача” описуються способи встановлення реалізованої системи на сервер, вимоги до технічної частини серверів та описана модель розробки програмного забезпечення.

У шостому розділі “Розробка стартап проекту” аналізуються варіанти просування запрограмованого веб-додатка, в якості стартап проекту.

# 1. ПОСТАНОВКА ЗАДАЧІ МАГІСТЕРСЬКОЇ ДИСЕРТАЦІЇ

Веб-система має бути представлена як ресурс, який надає можливість візуалізувати структуру KNX мережі та показати можливі способи взаємодії компонентів мережі.

Метою дослідження є реалізація програмного забезпечення інтерфейсу людина-машина для візуалізації структури мережі KNX та швидкого налаштування алгоритму взаємодії мережевих компонентів за допомогою інструментів Node-RED.

Оскільки KNX є одним із найпоширеніших рішень для середніх та великих систем автоматизації для будинків та, система повинна контролювати мережеві компоненти через Інтернет з будь-якої відстані [1].

Оскільки мережу KNX можна легко адаптувати до нових завдань та розширювати, а нові компоненти можна просто підключити до вже запущеної системи.

Важливою частиною програмного забезпечення є зберігання всіх подій, що відбуваються в мережі. Використовувана система графічного програмування Node-RED дозволяє переглядати історію використання системи та запити, а також визначати найкращі способи використання мережі KNX.

Система повинна мати можливість працювати з використанням модулів веб-системи Node-RED та тісно взаємодіяти з можливостями опису алгоритму роботи компонентів системи.

Програмний продукт повинен забезпечити наступні можливості:

- створення та налаштування особистих компонентів в системі Node-RED;
- створення та налаштування алгоритму взаємодії компонентів системи з використанням нових компонентів;
- реалізувати адаптер підключення Node-RED системи до KNX мережі;
- “спілкування” веб-системи з мережею;

- моніторинг стану компонентів мережі;
- інтерфейс для кінцевого користувача;
- інтерфейс для технічного спеціаліста Node-RED системи;

## **Висновки до розділу 1**

Основним завданням є розробка людино-машинного інтерфейсу для візуалізації KNX мережі та взаємодії компонентів мережі. Система повинна забезпечити можливість швидкого налаштування алгоритму взаємодії компонентів мережі використовуючи веб-інтерфейс Node-RED та можливість дистанційного управління компонентами мережі з будь-якого пристрою використовуючи мережу Інтернет.

## **2. СПОСОБИ РЕАЛІЗАЦІЇ ВІЗУАЛІЗАЦІЇ KNX МЕРЕЖ**

На сьогодні KNX - одне з найпоширеніших рішень у великих та середніх системах автоматизації офісів, комерційних приміщень та інших будівель. Вимоги на комфорт, безпеку та гнучкість сучасних будівель постійно зростає. Тому попит на інтелектуальні системи, які відповідають цим умовам, зростає.

Інтерфейс користувача - це зовнішня оболонка програмної системи, яка дозволяє користувачеві взаємодіяти з програмою; набір інструментів для обробки та відображення інформації, максимально адаптованих до комфорту користувача. У веб-системах користувацький інтерфейс реалізовується багато віконним режимом, можливістю налаштовувати загальний вигляд вікон, змінюючи колір, розмір та видимість користувацького вікна. Також важливе розташування вікон, сортування їх елементів, можливість гнучкого налаштування окремих елементів інтерфейсу користувача та доступність можливостей змінювати інтерфейс за вимогами кінцевого користувача.

### **2.1. Користувацький інтерфейс**

Користувацький інтерфейс (ІК) - це набір інструментів, використовуючи їх кінцевий користувач взаємодіє з будь-якими пристроями (комп'ютером чи побутовою технікою) або іншими вдосконаленими інструментами (системами). Користувацький інтерфейс - це різновид інтерфейсу, в якому взаємодіє та машина (пристрій, програмне забезпечення) [2]. Як визначено Національним банком стандартних науково-технічних концепцій, інтерфейс користувача - це набір апаратних та програмних засобів, що дозволяє користувачеві взаємодіяти з комп'ютером. ІК часто розуміють лише як появу програмного забезпечення (програмного забезпечення), але це розуміння занадто вузьке, оскільки саме через

інтерфейс користувач сприймає програму в цілому і використовує її функціональність.

ІК дозволяє користувачеві сприймати програмне забезпечення в цілому і використовувати його функціональні можливості. ІК надає підтримку у прийнятті рішень у певній тематичній області та визначає, як слід використовувати програмне забезпечення та його документацію. Фактично, ІК поєднує всі програмні компоненти та компоненти, які можуть впливати на взаємодію користувачів із програмним забезпеченням. До них належать: набір завдань, які користувач вирішує за допомогою програмного забезпечення; контроль; навігація між блоками програмного забезпечення; візуальне (і не тільки) оформлення вікон та форм заявок та інших компонентів (рисунок 2.1).

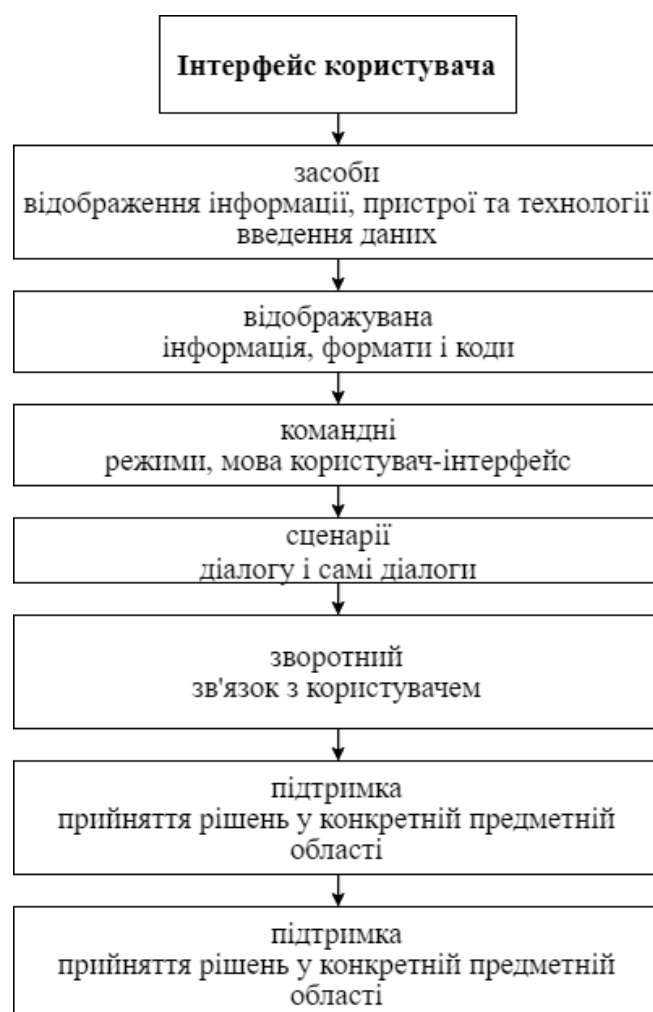


Рисунок 2.1 – Компоненти інтерфейсу користувача



Процес проектування інтерфейсу - це складний, нелінійний, недетермінований і неортогональний процес. Складність є загальною рисою програмного забезпечення (програмного забезпечення), але тим більше у випадку інфрачервоного зв'язку через безліч факторів та невизначеностей, які впливають на його розвиток. ІК-дизайн - це нелінійний процес, оскільки немає необхідності у фіксованій, упорядкованій та прямій лінії від початку до кінця.

## 2.2. Проблеми проектування інтерфейсів користувача

Процес проектування користувацького інтерфейсу став самостійною проблемою, яка часто переважає складність створення коду і, як і процес проектування будь-якої складної системи, вимагає відповідних методів, інструментів і, звичайно, зусиль кваліфікованих фахівців (рисунок 2.2).

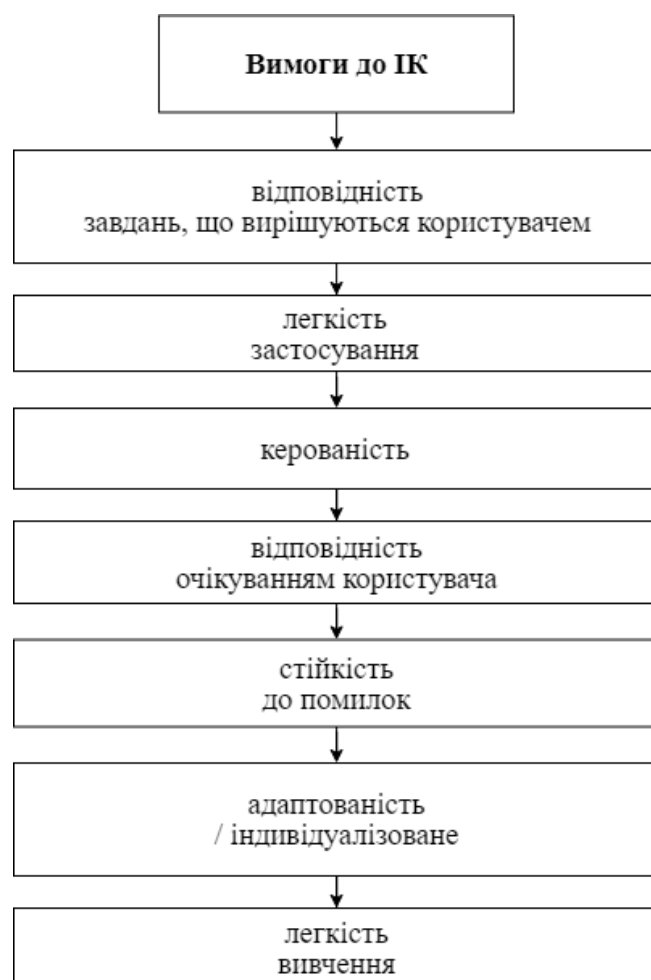


Рисунок 2.2. – Вимоги до інтерфейсу користувача

Розробники програмного забезпечення намагаються відокремити, хоча б концептуально, функціональність програмного забезпечення з точки зору проблем, пов'язаних з організацією взаємодії користувачів з цими продуктами. Зауважимо, що сама концепція дружби - це вектор, який містить, згідно з міжнародною класифікацією, сім вимог користувацького інтерфейсу:

### **2.3. Принципи проектування інтерфейсів користувача**

Інтерфейс користувача надає можливість для взаємодії користувача та комп'ютера (HCI - Human-Computer Interaction). Основна мета HCI - поліпшити взаємодію між користувачем та комп'ютером, зробивши комп'ютери більш корисними та адаптованими до потреб користувача.

Часто ефективність використання всіх системних функцій та продуктивність самої системи більше залежить від того, як побудований її інтерфейс [3].

Природний інтерфейс - це інтерфейс, який не вимагає від користувача істотної зміни звичних способів вирішення проблеми. Це означає, зокрема, що повідомлення та результати, отримані за допомогою програмного забезпечення, не потребують подальшого уточнення. Також доцільно підтримувати систему позначень та термінологію, яка використовується в даній місцевості.

Використання звичних для користувача понять та зображень (метафор) забезпечує інтуїтивний інтерфейс при виконанні завдань.

Метафори – це “місток”, що пов'язує об'єкти, якими доводиться користуватися користувачу в момент роботи на комп'ютері та образи, з яким пов'язаний реальний світ. Вони забезпечують “пізнавання”, а не саму “згадку”. Користувачі намагаються запам'ятати дію, яка пов'язана зі знайомим об'єктом. Це є набагато простіше, ніж запам'ятати назву команди, яка пов'язана з цією дією.

Послідовність інтерфейсу означає використання однакових або дуже схожих метафор та способів взаємодії з користувачем та порядку роботи в різних додатках. Послідовність дозволяє користувачам переносити наявні знання до нових завдань, швидше вивчати нові аспекти і, таким чином, зосереджувати свою увагу на вирішеній

задачі, замість того, щоб витрачати час на пошук відмінностей у використанні певних елементів управління, команд тощо. Забезпечуючи безперервність набутих раніше знань та навичок, послідовність що інтерфейс зрозумілий і передбачуваний.

Згода елементів є важливою для всіх варіантів інтерфейсу, а саме назви команд, візуальне відображення інформації та поведінку його елементів

У користувача можуть виникнути проблеми при роботі з деякими елементами системи, якщо дія програмного компоненту виходить за рамки того, що зазвичай здається йому відповідною метафорою на об'єкт у реальному світі. Якщо для програмного компоненту системи Робочий стіл назначити дію Запуск, то користувачу потрібна буде допомога для пошуку та вивчення значення цього об'єкта у системі.

Основним методом вивчення особливостей роботи нового програмного продукту є метод помилок та спроб. Кожен інтерфейс повинен мати на увазі конкретний підхід: на кожному етапі роботи слід передбачати лише відповідний набір дій та сповіщати користувачів про ситуації, в яких вони можуть завдати шкоди системі чи даним; ще краще, якщо користувач має можливість скасувати або виправити вжиті дії.

Гнучкість інтерфейсу - це здатність інтерфейсу користувача передбачати продуктивність користувача і рівень навчання користувача веб-системи. Властивість гнучкості допускає можливість налаштування конструкції діалогу та вхідних даних[3]. Концепція гнучкого інтерфейсу є одним із основних напрямків дослідження взаємодії людина-комп'ютер. Основна проблема полягає не в тому, як організувати зміни в діалозі, а які функції використовувати для визначення потреби в змінах та їх природи.

Проектування візуальних елементів є важливою частиною створення програмного інтерфейсу. Правильне візуальне подання використаних об'єктів забезпечує достатньо важливу додаткову інформацію про поведінку та взаємодію різних об'єктів. У той же час майте на увазі, що кожен візуальний елемент, який з'являється на екрані, потенційно потребує уваги користувача, який, як відомо, є нескінченним. Необхідно створити таке екранне середовище, яке не тільки допоможе

вам зрозуміти інформацію, надану користувачем, але і дозволить зосередитись на її найважливіших аспектах.

## **2.4. Комунікаційна шина KNX**

KNX - це технологія, яка забезпечує комунікаційну платформу для взаємодії різних пристроїв та систем. У 1990 році було створено асоціацію, що складається з 15 великих виробників приладів автоматики. Метою асоціації було просування самої технології, контроль якості та сумісності продукції, що виробляється компаніями-членами асоціації, та створення навчальних програм для працівників. Пізніше технології EIV, EHS (European Home System) та Batibus об'єдналися в KNX, і асоціація стала відома як Асоціація KNX, до якої зараз входить понад 350 компаній у всьому світі, а кількість пристроїв, підтримуваних цим протоколом, вже давно перевищила число 7000 (рисунок 2.3) [4].

В даний час KNX - одне з найпоширеніших рішень для використання в середніх і великих системах автоматизації в будинках, офісах та комерційних приміщеннях. Він з'явився на ринку більше двадцяти років тому і сьогодні підтримує багатьох великих виробників електрообладнання.

Сьогодні понад 350 компаній з усього світу приєдналися до KNX. Кількість сертифікованих товарів перевищує 7000. Орієнтовна кількість пристроїв, встановлених у великій кількості пристроїв, у відповідь на ці стандартні стандарти, оцінюється у понад десять мільйонів. Найвідоміші виробники пристроїв KNX сьогодні використовують ABB, Gira та Schneider Electric.

KNX - це шина зв'язку, яка широко використовується в автоматизації будівель. Стандарт шини KNX був розробкою найбільш ранньої версії EIB (аббр. Від англ. European Installation Bus, рус. Європейська інсталяційна шина)[4]. EIB - застаріла назва, але воно все ще використовується, особливо в Європі.. Також можна побачити позначення EIB / KNX. Продукція KNX набула популярності між 23 торговельними марками. Найвідоміші це GIRA, JUNG, INSTABUS, iBus, INTESIS BOX.



Рисунок 2.3 – Можливості комунікаційної шини KNX

Основні ключові особливості KNX - це гарантована сумісність продуктів різних виробників, один програмний інструмент (ETS) для планування, розробки та впровадження проектів, а також офіційні спеціалізовані курси навчання та сертифікації. Технічно рішення реалізують усі популярні сценарії автоматизації, включаючи освітлення, клімат-контроль та безпеку.

Багато проектів KNX засновані на використанні спеціальної шини з витвою парою. Всі контролери, датчики та пускачі приєднані до шини. На практиці це означає розробку дизайну та необхідне спілкування під час будівництва чи ремонту. Формально в стандартах є й інші трансляції, але вони відносно рідкісні в проектах. Крім того, він не вимагає встановлення додаткової шини управління, і часто схеми використовуються з усіма окремими користувачами на загальному екрані. Обидва способи мають свої переваги та недоліки. Також рекомендується їх поділити, якщо збережена специфікація KNX (рисунок 2.4).

## Schematic representation of the KNX bus

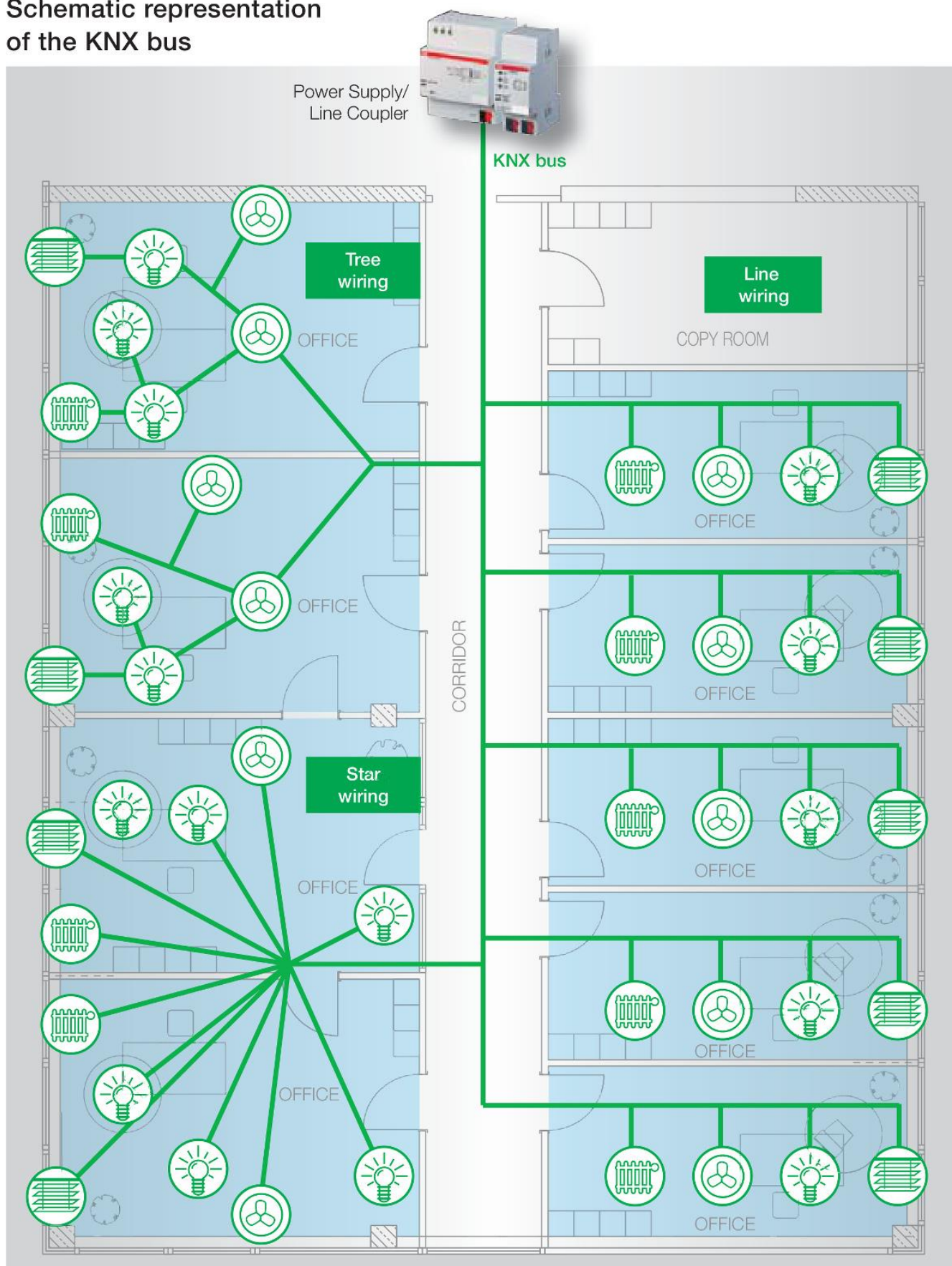


Рисунок 2.4 – Схематичне зображення комунікаційної шини KNX

Стандарт KNX був презентований весною 2002 року. Як виявилось, близько 80 відсотків основних змін були запозичені у EIB.

KNX отримав механізм завдання налаштувань та механізм передачі даних та сигналів від двох інших “донорів”. Ось чому KNX та EIB часто називають одним і тим самим, досить часто шині дають назву “EIB / KNX” [5].

В 2003 року новітня розробка була визначена за допомогою європейського стандарту EN50090. Через три роки розробка отримала статус міжнародного стандарту ISO/IEC14543. Як показує практика, сьогодні KNX мережі мають успіх у застосуванні не тільки у межах Європи.

Які переваги стандарту? Перш за все, вона відома своєю надійністю: незважаючи на наявність багатьох середовищ передачі даних, основні елементи систем KNX зазвичай з'єднуються між собою за допомогою спеціальних кабелів, а система забезпечує механізм підтвердження отримання пакетів, тобто якщо команда не досягла цілі, то її знову відправляють (не більше двох разів). Ще один очевидний плюс провідної мережі це можливість розташування обладнання на вагомій відстані один від одного. Слід також зазначити, що пристрої KNX не мають проблем із сумісністю, що не стосується, наприклад, конкурентних продуктів. Гнучка масштабованість - ще один аргумент для KNX. Стандарт може однаково ефективно застосовуватися в приватних будинках, а також у лікарнях чи аеропортах. Недоліками KNX також є: цей продукт орієнтований на професійні системи автоматизації, які розроблені та встановлені компаніями, що інтегруються. Незалежна установка мережі KNX - справжнє завдання.

Компоненти мережі KNX можна поділити на три основні групи. Перша група складається з пристроїв, що створюють керуючі команди. Це можуть бути вимикачі, пульти управління, різні датчики та таймери, а також датчики для вимірювання фізичних величин.

Друга категорія описує актуатори, тобто пристрої які виконують саму роботу в мережі. В другу категорію можна віднести різні регулятори та релейні модулі. Яскравим прикладом другої групи є перемикачі.

Третя група складається з допоміжних системних пристроїв системи. Наприклад, блок живлення, логічний модуль, повторювач та інтерфейси, що надають зв'язок з мережею Інтернет. Мережа KNX має характеристику децентралізованої мережі. Тобто сенсори і модулі другої категорії мають можливість обмінюватися повідомленнями безпосередньо, не використовуючи додатковий контролера (Рисунок 2.4).



Рисунок 2.4 – Головний контролер для підключення до KNX-мережі

Стандарт KNX охоплює чотири середовища передачі даних: окрему шину (кручена пара), кабельне з'єднання, радіоканал та мережу IP. Їх не можна назвати



однаковими. Зв'язок по шині дозволяє використовувати різні варіанти топології мережі та підключати велику кількість пристроїв на значній відстані один від одного.

У самому простому варіанті мережа KNX являє собою сегмент з “лінійною” топологією. Він може містити до 64 пристроїв шини. Максимальна довжина однієї лінії - один кілометр, але за допомогою спеціальних підсилювачів значення можна збільшити в чотири рази. Кожен сегмент повинен мати власне джерело живлення. До 15 ліній можна підключити до так званої основної лінії і таким чином з'єднати в "зону". У свою чергу, зони (до 15 штук) здатні спілкуватися один з одним по міській лінії.

Швидкість передачі даних в KNX мережі становить 9600 біт / с, але через низький рівень широкомовного повідомлення (кілька байт), цього достатньо для забезпечення хорошої відповіді: середній час відповіді на команду становить лише 25 мс. Пакети, що надсилаються, мають пріоритет. Слід зазначити, що така мережа використовує два типи адрес - фізичну та групову. Остання опція використовується, коли потрібно одночасно надіслати одну команду на декілька пристроїв - вона визначає приналежність гаджета до певної умовної групи. Фізична адреса мережевого елемента завжди одна, але адреса групи може бути декількома.

Бездротова версія KNX мережі використовує свою частоту 868 МГц, в той же час не більше одного відсотка ефірного часу відводиться для передачі сигналу від окремих пристроїв, що дозволяє уникнути тривалої перешкоди, що блокує радіоканал.

Максимальна швидкість передачі даних приблизно 16400 біт за секунду. Однонаправлені гаджети негайно надсилають пакети, двонаправлені параметри попередньо перевіряють, чи є вільний радіоканал. Повідомлення в ефірі включають, але не обмежуються цим, рівень акумулятора та серійний номер пристрою. Останнє дозволяє уникнути проблем при використанні декількох радіомереж в одній області. Радіус діапазону KNX порівняний з діапазоном його прямих конкурентів, тоді як діапазон підсилювача збільшується завдяки спеціальним ретрансляторам.

Можливість використання електрокабелів як носія передбачена у випадках, коли важко прокласти новий кабель і радіосигнал не поширюється в достатній мірі.

Технологія дозволяє домогтися швидкості близько 1200 біт за секунду. Слід зазначити, що цей варіант не є популярним, і в практичних реалізаціях він зустрічається рідко. У такому середовищі, як IP-мережа, він використовується для тунелю та маршрутизації команд KNX шляхом їх перетворення в IP-пакети. Ця функція особливо корисна, коли інтерактивні пристрої розташовані на значній відстані один від одного.

Пристрої KNX недостатньо просто об'єднати між собою - їх після цього необхідно ще й налаштувати. До недавнього часу стандарт передбачав три режими конфігурації системи - A-mode (Auto), E-mode (Easy) і S-mode (System). Перший варіант - найпростіший і створювався він як раз для новачків, які займаються самостійним монтажем. Передбачалося, що A-mode дозволить автоматично об'єднувати пристрої, функції яких мають однозначну прив'язку один до одного. Єдині дії з боку користувача - підключення гаджетів до середовища передачі даних і вибір визначених параметрів. На жаль, A-mode «не злетів», і від нього, в кінцевому рахунку, відмовилися.

Режим E-mode - варіант, орієнтований на невеликі системи автоматизації. При його використанні настройка компонентів здійснюється за допомогою кнопок, розташованих на самих гаджетах, або за допомогою центрального контролера, причому останній, як правило, може бути видалений з системи після завершення конфігурації. В цілому, можливості E-mode обмежені, але їх, нерідко, виявляється досить для завершення пусконаладжувальних робіт.

Системний режим відкриває доступ до налаштувань всіх параметрів пристроїв, ось тільки для використання S-mode потрібне спеціальне програмне забезпечення - ETS (Engineering Tool Software). Даний продукт є універсальним, і, теоретично, він сумісний з усіма високотехнологічними KNX-гаджетами. Програма дозволяє підключатися до зовнішніх інтерфейсів мережі автоматизації за допомогою комп'ютера, використовуючи RS-232 (COM-порт), USB або LAN. Якщо KNX-система має доступ в інтернет, то контроль над її роботою може здійснюватися віддалено. Слід зазначити, що ETS активно використовується не тільки для настройки, а й для проектування мережі.

## 2.5. Веб-система Node-RED

Для мінімальної реалізації системи, яка взаємодіє з KNX мережею, потрібно чимало технічних знань в сфері розробки програмного забезпечення та налаштування мережі між компонентами системи. Також потрібно мати великий досвід роботи з KNX мережею та її компонентами. KNX компоненти не мають свого веб-інтерфейсу для взаємодії між ними та налаштування алгоритму взаємодії між ними. Основною проблемою розробки інтерфейсу взаємодії з KNX мережею було полегшити її налаштування, прибрати неважливі технічні деталі при налаштуванні алгоритму взаємодії її компонентів та створення графічного відображення взаємодії компонентів мережі.

За основу розробки візуалізації інтерфейсу взаємодії компонентів KNX мережі було взято систему Node-RED. Ця система дає можливість графічно відобразити представлення зв'язків та алгоритмів взаємодії компонентів, надає простоту в програмуванні поведінки компонентів мережі та швидкість налагодження правильної роботи системи та зручність обміну специфічного алгоритму взаємодії компонентів між KNX мережами.

Платформа Node-RED надає користувачам та розробникам багатий набір інструментів розробки робочих процесів, а також сервіси та протоколи взаємодії між компонентами, що забезпечують всебічну підтримку повного життєвого циклу робочих процесів [6].

Програмування здійснюється в браузерному редакторі, де користувачі можуть створювати потоки, підключаючи один до одного вузли різного призначення, а також виконувати розгортку створених потоків в середу виконання всього в один клік (Рисунок 2.5).

Також в наявності текстовий редактор, що дозволяє створювати JavaScript-функції прямо в редакторі Node-RED.

Вбудована бібліотека дозволяє зберігати корисні функції, шаблони і потоки для повторного використання.

На даний момент в сховище пакетів Node.js числиться понад 225 тисяч модулів, що дає широкий простір для розширення можливостей проекту за допомогою нових вузлів.

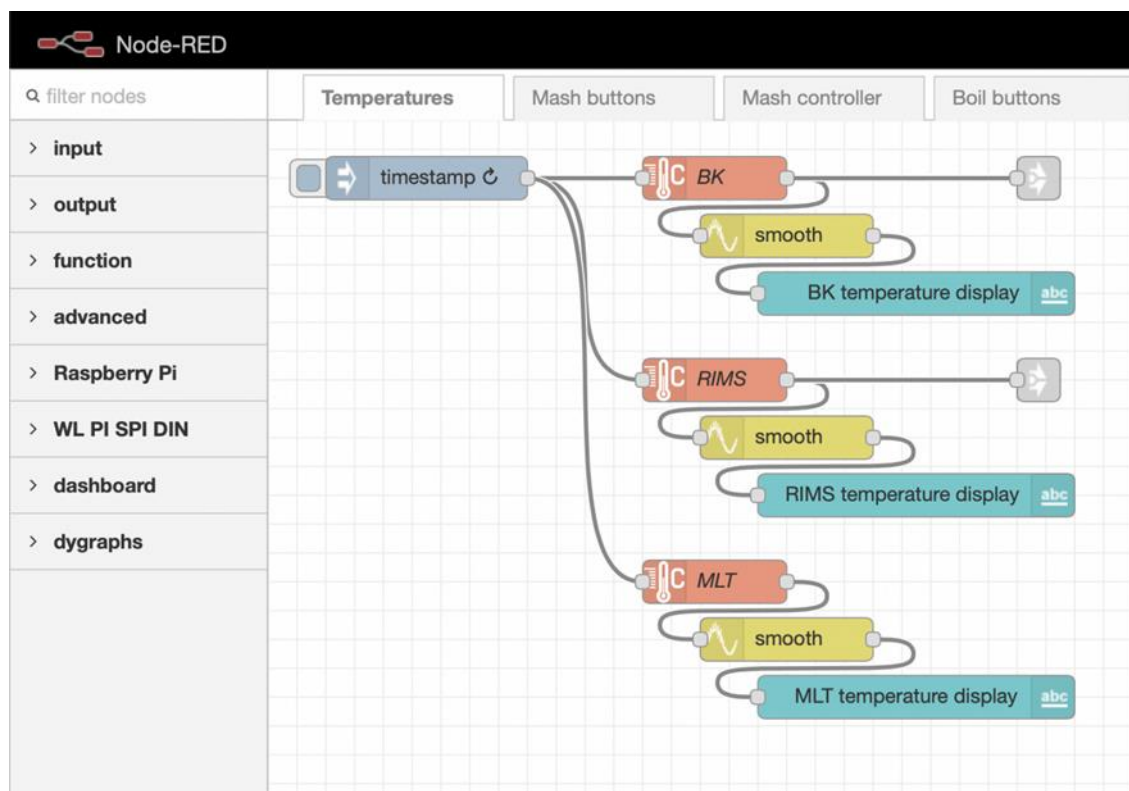


Рисунок 2.5. – Веб-редактор Node-RED

Node-RED розроблена на базі Node.js і завдяки цьому максимально використовує його подієво-орієнтовану, неблокуючу модель. Це робить Node-RED ідеальним додатком для запуску на мережевий периферії - на недорогих пристроях Raspberry Pi, KNX, а також в «хмарі».

## Висновки до розділу 2

Другий розділ визначає поняття інтерфейсу користувача, вивчає питання побудови інтерфейсу, визначає основні проблеми побудови користувацького

інтерфейса та описує основні принципи дизайну інтерфейсу користувача. Визначено поняття мережі KNX, її структуру та особливості. Наведено також приклад підключення елементів до мережі KNX та без неї. Розглянуто головний інструмент Node-RED для підключення, налаштування та підключення до KNX-мережі. Розглянуто головні можливості для створення особистого алгоритму взаємодії компонентів мережі з використанням веб-додатку Node-RED.

### **3. ЗАГАЛЬНИЙ ОПИС ІНСТРУМЕНТІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

При розробці програмного забезпечення важливим фактором є правильний вибір реалізації програмного забезпечення, який впливає на час розробки, якість, надійність та швидкість продукту. Тому JetBrains, мова програмування JavaScript, була використана для розробки програмного забезпечення. Node.js та онлайн-система візуального програмування Node-RED використовувались для впровадження програмного забезпечення на стороні сервера. JQuery, HTML для верстки веб-сайту та CSS для опису веб-сайту використовувались для розробки клієнтської частини продукту. Для створення деяких деталей інтерфейсу було використано Adobe Photoshop CS6, програма для редагування, моделювання та створення графічних елементів.

#### **3.1. Середовище розробки WebStorm JetBrains**

Програмне середовище JetBrains WebStorm - це інструмент для створення веб-сайтів та редагування HTML, CSS та JavaScript. Рішення забезпечує швидку навігацію по файлах та генерує проблеми з кодом у режимі реального часу. JetBrains WebStorm дозволяє вам додавати теги HTML або елементи SQL безпосередньо в JavaScript. JetBrains WebStorm реалізує та синхронізує проекти через FTP [7].

Використовуючи можливості HTML / XHTML та XML, WebStorm автоматично завершує стилі, посилання, атрибути та інші елементи коду. Під час роботи з кодом класу CSS, номерами HTML, ключовими словами тощо. Повний WebStorm забезпечує автоматичне вирішення таких проблем, як вибір формату, властивості, класи, посилання на файли та інші атрибути CSS. Рішення дозволяє використовувати потужність інструменту кодування Zen для розмітки HTML, відображає активність тегів на веб-сторінці. Продукт WebStorm доповнює JavaScript ключовими словами,

мітками, змінними, параметрами та функціями DOM та підтримує конкретні функції популярних браузерів. Функції рефакторингу JavaScript, реалізовані в рішенні, дозволяють перетворити структуру коду, файли та .js файли.

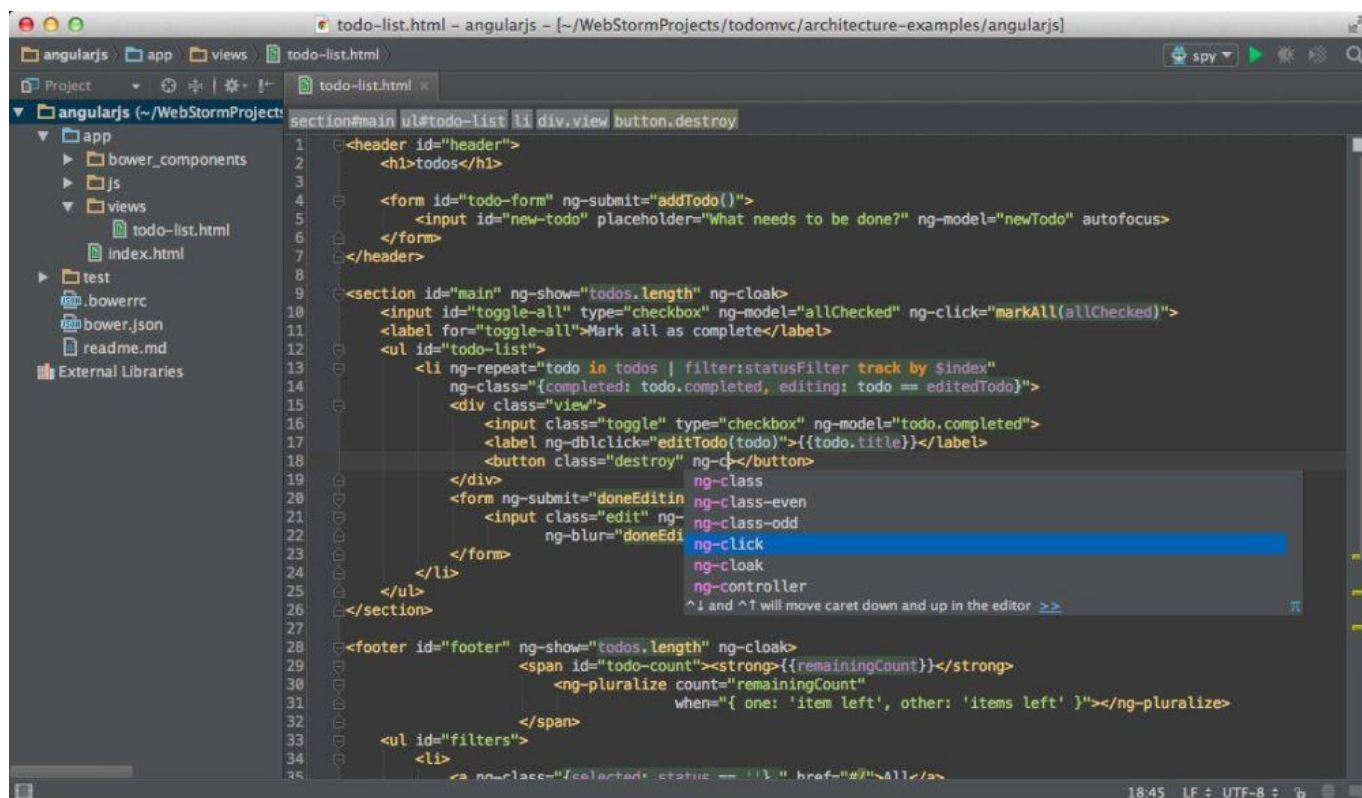


Рисунок 3.1. – Загальний вигляд середовища розробки WebStorm

WebStorm надає налагодження JavaScript та пропонує широкий спектр функцій: пошук точок перерви в HTML та JavaScript, налаштування точок зупинки, тестування синтаксису коду в реальному часі тощо. Продукт підтримує JQuery, YUI, Prototype, DoJo, MooTools, Qooxdoo та Bindows. WebStorm забезпечує інтегровану перевірку тексту для тегів, послідовностей кодів, написання помилок і т. Д. WebStorm дозволяє редагувати файли та автоматично синхронізувати їх на вимогу під час віддаленої роботи або зберігання.

WebStorm, як і інші IDE, засновані на платформі IntelliJ IDEA, робить створення простішим та зручнішим, забезпечуючи виділення та автоматичне завершення коду, його аналіз у режимі реального часу, швидку навігацію та рефакторинг, а також забезпечує програмістів потужною налагодженням та інтеграція із системами

управління версіями. WebStorm дійсно розуміє структуру та код проекту, визначає можливі проблеми перед відкриттям проекту в браузері та пропонує рішення. Вбудовані інструменти IDE для тестування та управління проектами допоможуть вам розробити та зробити його більш зручним та ефективним.

Редагування в реальному часі дозволяє одразу побачити результати змін у файлах CSS, HTML та JavaScript проекту в браузері без перезавантаження сторінки. Живе редагування доступне в режимі налагодження JavaScript і працює в Google Chrome із попередньо встановленою підтримкою JetBrains IDE.

WebStorm дозволяє ефективно створювати програми на Node.js та підтримує повну налагодження програм Node.js. Ви можете створити нову програму за допомогою шаблону Node.js Express та встановити необхідні модулі за допомогою вбудованого менеджера npm WebStorm.

Ключові можливості:

- Live Edit;
- налагодження коду на JavaScript, а також на ECMAScript 2015 року, TypeScript, CoffeeScript і Dart з використанням source maps;
- налагодження Node.js додатків;
- можливість взаємодії з системою управління версіями Git, Subversion та Mercurial;
- тісна взаємодія з системами стеження за помилками ESLint та JSHint;
- повна підтримка Angular, React, jQuery, Electron та іншими додатками.

Продукт підтримує функцію контролю версій і попередніх варіантів коду і фіксує всі вироблені дії та зміни. Завдяки створенню історії в WebStorm можна відновлювати кодові вирази, блоки і навіть цілі файли.

## 3.2. Мова програмування JavaScript

JavaScript - це динамічна, об'єктно-орієнтована мова програмування. Впровадження стандарту ECMAScript. Найчастіше використовується як частина браузера, що дозволяє клієнтському коду (виконуваному на пристрої кінцевого



користувача) взаємодіяти з користувачем, керувати браузером, асинхронним зв'язком із сервером, змінювати структуру та зовнішній вигляд веб-сайту (рисунок 3.2) [9].

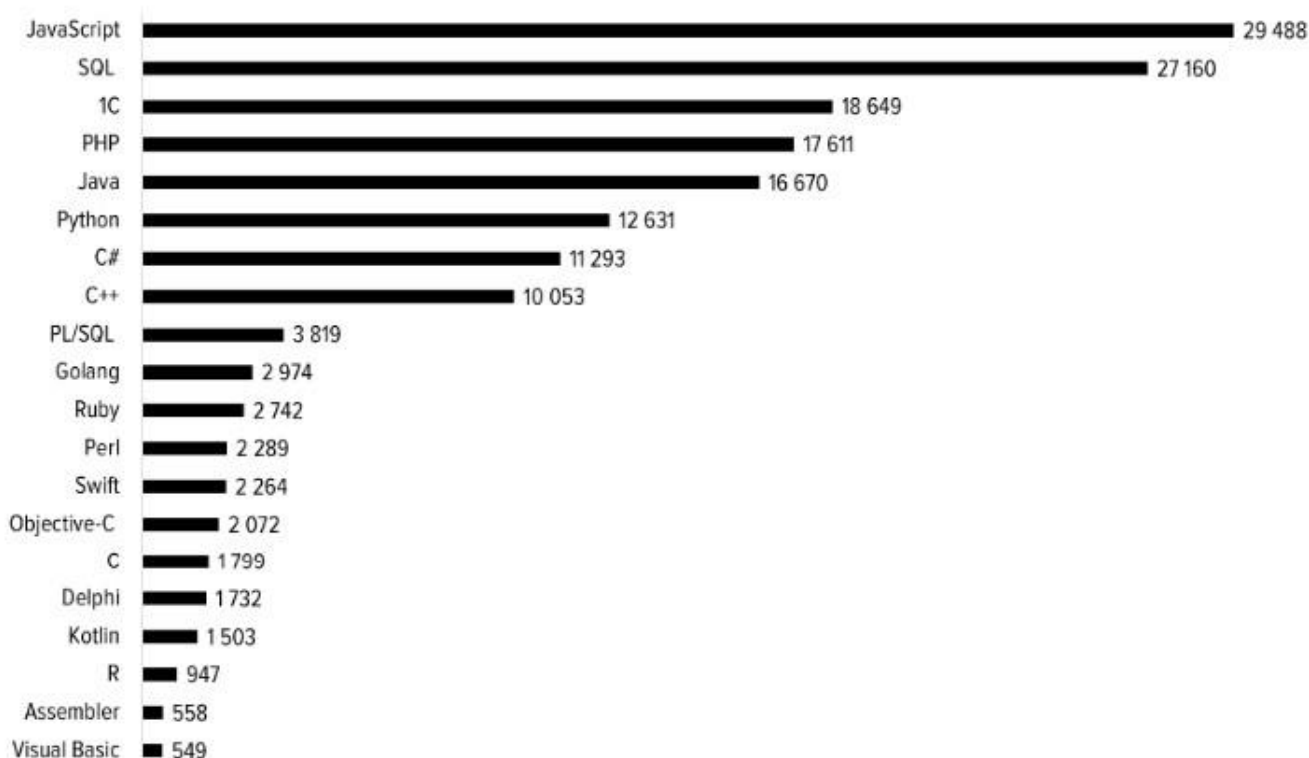


Рисунок 3.2 – Рейтинг мов програмування у 2019 р. за версією dou.ua

JavaScript - це об'єктно-орієнтована мова програмування для створення скриптів як на клієнтській, так і на стороні сервера. Тому це не «повноцінна» мова програмування, а скоріше зосереджена на використанні можливостей сценарію середовища.

Клієнтський браузер забезпечує середовище, в якому JavaScript має доступ до таких об'єктів, як вікна, меню, діалогові вікна, текстові області, рамки, файли cookie та веб-сайт I / O. Крім того, браузер дозволяє приєднувати JavaScript до таких подій, як завантаження та вивантаження сторінок і графіки, натискання клавіш і рухи миші, підбір тексту та подання форм. У цьому випадку код скрипту реагує лише на події і тому не потребує основної програми.

Набір об'єктів, наданих браузером, відомий як Документна модель об'єкта (DOM). Веб-сервер забезпечує інше середовище, в якому об'єктами є запити до бази

даних, клієнти, файли та механізми блокування та обміну. Спільний доступ до сценаріїв на стороні клієнта та сервера дозволяє розподіляти обчислення між ними та надавати дружній інтерфейс для веб-додатків.

JavaScript - це сукупність інтерактивних об'єктів. Об'єкт JavaScript - це нерегульований набір властивостей, кожен з яких має нульовий або більше атрибутів, які визначають спосіб використання цього властивості.

Програми JavaScript є автономними та вбудовані в документи HTML. JavaScript інтерпретується самим браузером при завантаженні документа, що містить його код. Програми JavaScript підтримують різноманітні завдання і можуть бути настільки ж складними (або простими), як потрібно. Колись вираз "HTML програмування" викликав хвилю критики серед програмістів, що працюють в Інтернеті. Але тепер HTML-документ може дійсно містити значну "програмовану" частину.

### 3.3. Платформа Node.js

Використання Node.js для написання API має багато вагомих переваг. Одним із найважливіших є використання неблокуючої моделі вводу та виводу (рисунок 3.3).

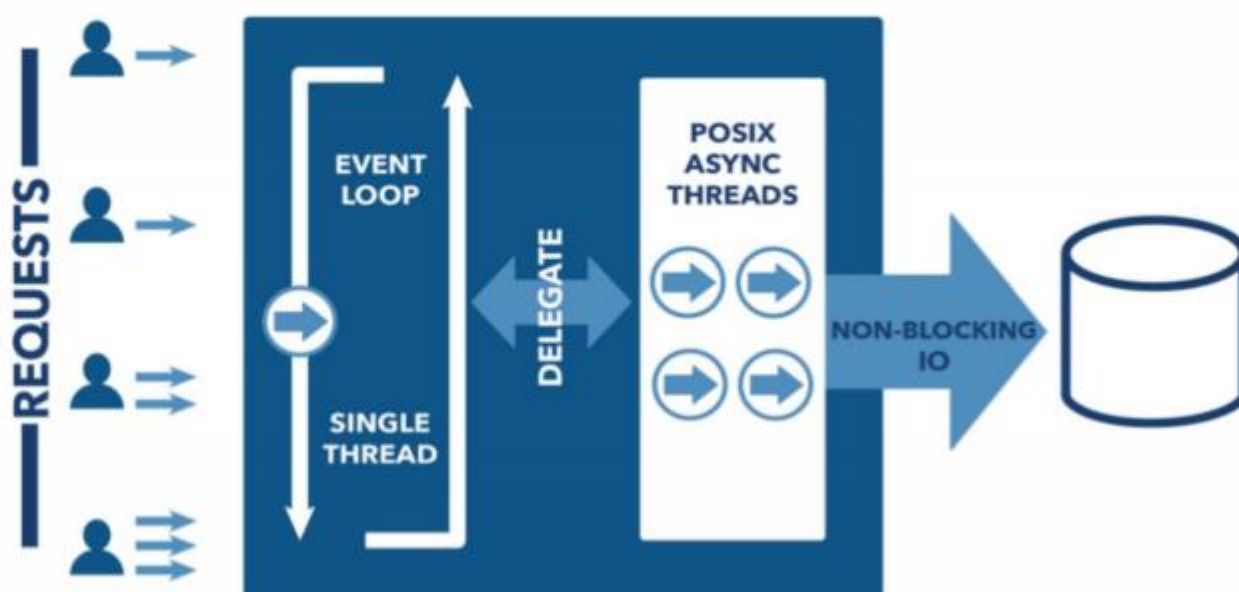


Рисунок 3.3 – Загальний принцип роботи неблокуючої моделі введення/виведення

Ця модель дозволяє обробляти велику кількість з'єднань одночасно, що має величезний вплив на масштабування системи. Це контрастує з популярною моделлю з використанням паралельних потоків операційної системи. Завдяки використанню Node.js вам не доведеться турбуватися про можливі процеси блокування, оскільки функції не працюють з пристроями вводу/виводу [9].

Розглянемо детальніше асинхронне введення та виведення (рисунок 3.4). Node.js використовує один потік, і цикл подій несе відповідальність за всі асинхронні введення-виведення. Поведінка програми регулюється функціями асинхронного зворотного виклику. Такий підхід дозволяє уникнути проблем, пов'язаних зі створенням та блокуванням окремих ниток. Крім того, це забезпечує значне збільшення швидкості відповіді сервера та масштабованості системи.



Рисунок 3.4 - Організація асинхронного введення/виведення у Node.js

Використання JavaScript у якості серверної мови дає значний приріст у швидкості виконання коду. Більше того, так як рушій V8 перетворює код JavaScript прямо у машинний код, це робить Node.js чи на найшвидшою платформу.

Оскільки Node.js заснований на архітектурі управління подіями, найкраще створювати програми в режимі реального часу. Оскільки і сервер, і сторінки клієнтів

написані на JavaScript, процес синхронізації між ними відбувається набагато швидше, простіше і зручніше.

Окрім вбудованого функціоналу, який постачає сам Node.js, в його склад входить також менеджер модулів npm (node package manager). Його основна ідея – це створення маленького програмного блоку, який вирішує одну конкретну проблему, але робить це швидко та якісно. Це дає можливість компонувати власні великі проекти з цих маленьких побудованих блоків. До того ж, це задовольняє принцип відкритості та багаторазового використання коду.

Слід зазначити, що Node.js повністю підтримує архітектурний шаблон програмування MVC (model-view-controller), який заслужено вважається загальноприйнятим стандартом.

З самої назви платформи «node» - вузол, можна побачити, що вона акцентує увагу на можливість побудови систем із великої кількості невеликих розподілених обчислювальних вузлів, які мають можливість обмінюватися даними між собою. Сервер на Node.js ніколи не виявиться потужнішим, ніж потрібно. Вся суть архітектури Node - у її мінімалізмі. При цьому серверну частину додатку можна масштабувати в залежності від потреб проекту.

Модульність Node.js дає можливість створювати невеликі додатки без необхідності підтримки величезної інфраструктури, більша частина функціоналу якої в результаті виявиться незадіяною. При розробці додатку на Node.js розробник сам може вирішувати, що йому потрібно, і, за необхідності, розширити проект.

### **3.4. Веб-платформа Node-RED**

Node-RED - це інструмент програмування візуального потоку, розроблений співробітниками IBM для поєднання різних пристроїв, API та онлайн-сервісів як частини Інтернету речей [10].

За допомогою Node-RED ви можете працювати з браузером редактора потоків даних як окремий вузол з різними функціональними можливостями, щоб можна було створювати функції JavaScript. Ви можете використовувати базові вузли, одразу

надані Node-RED, а також встановити вузли з додатковими функціями з сховища NPM або навіть створити власний вузол з унікальними функціями. Node-RED програми або їх частини можуть зберігатися та розповсюджуватися для вільного використання. Саме середовище базується на Node.js. Нитки, створені Node-RED, зберігаються як JSON. З MQTT версії 0.14, вузли можуть бути налаштовані для TLS-з'єднань.

Як було зазначено раніше, одним з найбільш важливих факторів обмежують розвиток інтернету речей є відсутність зручних засобів розробки правил взаємодії пристроїв IoT між собою. Для вирішення цього завдання був розроблений Node-RED, що дозволяє через браузер побудувати схему взаємодії пристроїв між собою і зовнішніми системами.

Дане рішення зручно як проміжне для зв'язку пристроїв різного типу між собою і / або ж з системою автоматизації або, наприклад, СУБД або іншим хмарою. З використанням додаткових пакетів Node-RED можна використовувати для створення простих систем автоматизації розумного будинку, але рішення буде відносно обмежених зважаючи неповного покриття функціональних потреб розумного будинку.

Node-RED працює на Node.JS, і був розроблений для роботи на відносно малопродуктивних системах.

З урахуванням озвучених факторів Node-RED зручно використовувати на шлюзах між різними мережами пристроїв інтернету речей функціонують на власних, як правило, більш простих протоколах і традиційним інтернетом, побудованих на TCP / IP, UDP. У цьому випадку він дозволить більш оптимально використовувати вільні ресурси шлюзу, що працює, як правило, на Linux.

Розглянемо практичну реалізацію на базі прикладу обробки посилки в TCP порт. Після запуску сервера графічний інтерфейс буде доступ по <http://localhost:1880/>. Інтерфейс гранично простий - по лівому краю наведені доступні блоки, які можна перетягнути в центральне поле і зв'язати їх, праворуч - консоль налагодження і виведення допоміжної інформації. Зазначу, що працює пошук за доступними блокам, що значно спрощує вибір необхідного блоку (рисунк 3.5).

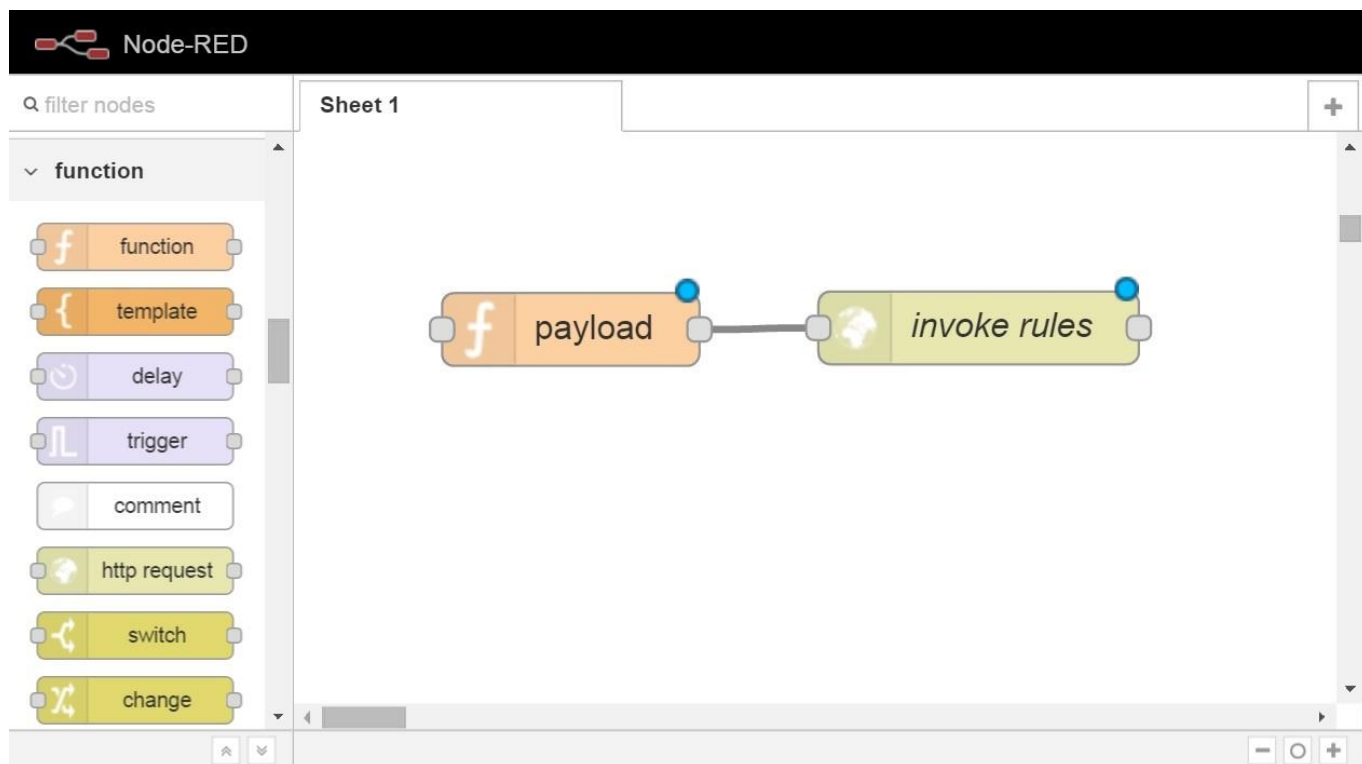


Рисунок 3.5 – Інтерфейс системи Node-RED

Після того як буде зібрана схема її необхідно розгорнути для виконання на сервері. Для цього використовується відповідна кнопка Deploy. (рисунок 3.6)

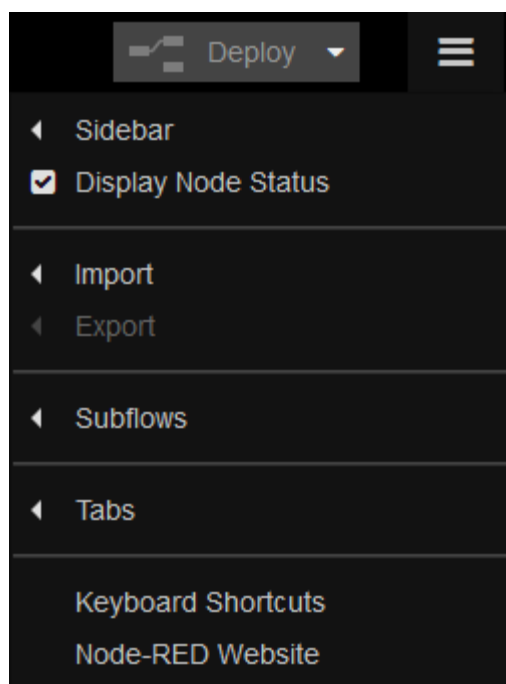


Рисунок 3.6 – Інтерфейс запуску алгоритму взаємодії компонентів системи Node-RED

Node-RED - це програма для програмування для з'єднання пристроїв, API та онлайн-служб.

Він містить браузерний редактор, з підтримкою якого легко з'єднати потоки, використовуючи різноманітні узори з набори, розвівшись у середі виконання однакових нажатих кнопок миші.

Node-RED - це міст між UiPath та цифровим миром. Він використовує Orchestrator, завдяки якому можливо підключити UiPath таким чином, як веб-сигнали, користувацькі інтерфейси та Інтернет-речі.

### 3.5. Бібліотека-розширення jQuery

jQuery - це бібліотека JavaScript, яка фокусується на взаємодії JavaScript та HTML. Бібліотека JQuery полегшує доступ, маніпулювання та доступ до будь-якого елемента DOM, доступ до атрибутів і вмісту DOM. Бібліотека jQuery також надає корисний API для роботи з AJAX (рисунок 3.7) [8].

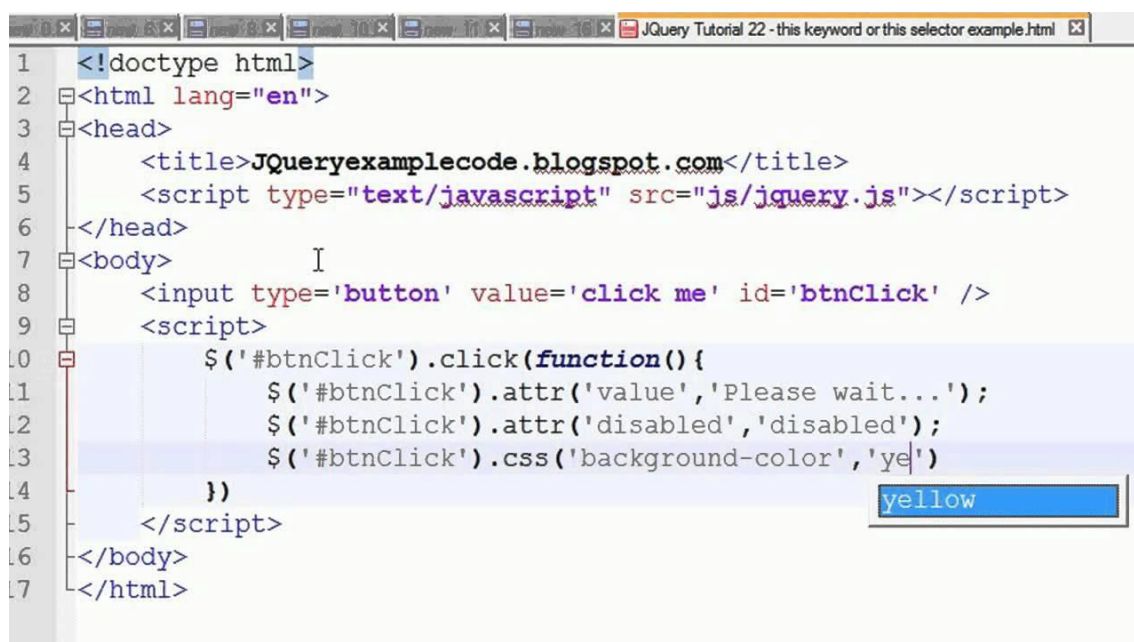


Рисунок 3.7 – Загальний вигляд можливостей jQuery

Синтаксис jQuery розроблений для полегшення навігації шляхом вибору елементів DOM, створення анімації, обробки подій та розробки програм AJAX.

jQuery також надає розробникам можливість створювати плагіни у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, програмісти можуть створювати абстракції для взаємодії низького рівня та створювати анімації для ефектів високого рівня. Це сприяє створенню потужних та динамічних веб-сайтів.

Основне завдання JQuery - надати розробнику простий та гнучкий інструмент для адреси DOM-об'єктів у браузері за допомогою селекторів CSS та XPath. Ця структура також пропонує інтерфейси для програм Ajax, обробників подій та просту анімацію.

Принцип jQuery полягає у використанні класу (функції), який повертається до себе під час доступу до нього. Це дозволяє побудувати цілісний ланцюжок методів.

### **3.6. Розмітка веб-сторінки HTML**

Документ гіпертексту - це текстовий файл із спеціальними тегами, звані тегами, які пізніше розпізнаються браузером і використовуються для відображення вмісту файлу на екрані комп'ютера. Ці теги дозволяють виділити заголовки документів, змінити колір, розмір шрифту та написання, вставити графічні зображення та таблиці. Але головною перевагою гіпертексту над простим текстом є можливість додавання гіперпосилань до вмісту документа - спеціальних конструкцій HTML, які дозволяють клацнути, щоб переглянути інший документ.

HTML - це стандартна мова розмітки для веб-сайтів. Більшість веб-сайтів створюються за допомогою HTML [12]. HTML-документ обробляється браузером і відображається на екрані аналогічно людському. HTML - це мова, похідна від SGML, успадковує від неї визначення типу документа та ідеологію структурних маркерів. Хоча HTML - це штучна комп'ютерна мова, вона не є мовою програмування. HTML, CSS та - це основні технології для створення веб-сайтів (рисунок 3.8).

HTML надає засоби для:

- розробки власного структурованого документу
- створення текстів різних типів, а саме заголовки, списки, цитати, таблиці, абзаци, тощо;



- одержання інформації з мережі Інтернет через гіперпосилання;
- створення форм взаємодії;
- додавання медіа.

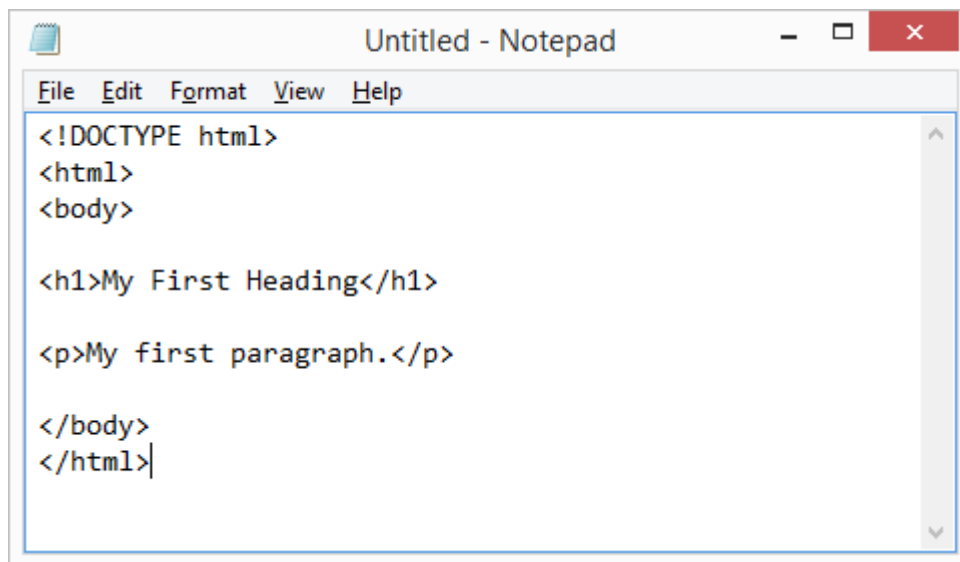


Рисунок 3.7 – Загальний вигляд сторінки HTML

### 3.7. Таблиці стилів CSS

Каскадні таблиці стилів - це мова, що застосовується для опису сторінок, написаних за допомогою мов розмітки даних (наприклад HTML). Найчастіше CSS використовується для візуального представлення сторінок, написаних у HTML та XHTML, але формат CSS може використовуватися для інших типів XML-документів (рисунок 3.9) [13].

Специфікації CSS були розроблені та розроблені консорціумом World Wide Web. CSS має різні рівні і профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нові функціональні можливості або розширюючи існуючі функції. Рівні позначені як CSS1, CSS2 та CSS3.

Профілі - це набір правил CSS одного або декількох рівнів, створених для конкретних типів пристроїв або інтерфейсів. Наприклад, існують CSS-профілі для принтерів, мобільних пристроїв тощо [16].

```

h1 {
    font-size: .75em;
    position: absolute;
    bottom: 20px;
    width: 94%;
    left: 2%;
}

h1:before {
    content: "This page is about: ";
    font-style: italic;
    display: inline;
}

main figure img {
    border: 5px solid #CCC;
}

.zebra-table {
    border: 1px solid #CCC;
    margin-bottom: 40px;
    margin: 0 auto;
}

.zebra-table th {
    text-align: center;

```

Рисунок 3.9 – Загальний вигляд можливостей CSS

CSS (каскадний або блоковий макет) замінив макет веб-електронних таблиць. Основна перевага компонування блоку - це відокремлення вмісту сторінки (даних) та їх візуальне представлення.

CSS використовується веб-авторами та відвідувачами для визначення кольорів, шрифтів, макета та інших аспектів зовнішності сторінки.

Однією з головних переваг є можливість відокремлення вмісту сторінки (або контенту, контенту, як правило, HTML, XML або подібної мови розмітки) від макета документа (описаного в CSS).

Один і той же документ HTML або XML може відображатися по-різному в залежності від використовуваного CSS. Основні переваги CSS - інформація про стиль для всього.

## Висновки до розділу 3

Для виконання цього завдання були використані сучасні технології, які є одними з найпопулярніших у світі, що забезпечують зручну підтримку дизайну та дозволяють легко масштабувати. Платформа Node.js та система графічного програмування Node-RED були використані для розробки сторінки сервера проекту, яка разом забезпечує швидку обробку даних, просте зберігання та доступ до даних. Node-RED та бібліотека JavaScript jQuery використовувались для реалізації інтерфейсів користувача, полегшуючи зв'язок із сервером та відображаючи дані користувачів. Описаний веб-сайт із тегом HTML та CSS.

## 4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Розробка програмного продукту поділяється на три частини: програмування адаптеру між Node-RED системою та KNX мережею, конфігурація власного алгоритму взаємодії компонентів мережі та створення веб-клієнта для кінцевого користувача. Веб-система Node-RED дає змогу створювати свої вузли для конфігурації особистого алгоритму взаємодії компонентів мережі. Адаптер був розроблений за допомогою платформи Node.js, мови розмітки HTML та стилів CSS. Клієнт взаємодіє з сервером за допомогою API REST, що дозволяє легко масштабувати клієнтські частини.

### 4.1. Адаптер між Node-RED та KNX мережею

Адаптера для роботи KNX-мережі з Node-RED системою імплементований за допомогою мови програмування JavaScript, платформи Node.js та мови розмітки HTML.

Адаптер для Node-RED складається з HTML файлу knx.html та knx.js компоненту. В knx.html описується загальний вигляд нових вузлів для системи Node-RED та файл knx.js описує функціональні можливості та алгоритм роботи нових вузлів.

Розроблений адаптер додає до існуючих стандартних вузлів веб-системи для графічного програмування Node-RED три нових вузла для взаємодії системи з KNX-мережею:

- “knx-controller”;
- “knx-in”;
- “knx-out”;

Створений вузол “knx-controller” для системи Node-RED не має графічного представлення і виконує функцію конфігурації для з’єднання з KNX мережею для інших двох вузлів. Цей вузол є обов’язковим для з’єднання веб-системи Node-RED з

KNX-мережею. Для налаштування вузла потрібно вказати адресу, порт та тип з'єднання до контролера KNX-мережі.

Усі запити до мережі мають спеціальний формат, встановлений KNX мережею та відправляються у форматі JSON. Поля, описані нижче, є обов'язковими для заповнення для коректної роботи системи та передачі даних до певного компонента мережі (рисунок 4.1).

```
{
  "topic": "write",
  "payload": {
    "dstgad": "0/1/0",
    "dpt": "1",
    "value": 0
  }
}
```

Рисунок 4.1 – Загальний вигляд повідомлення до KNX мережі

Вузол “knx-out” використовує стандартні вузли веб-системи Node-RED, конвертує вхідні дані до стандартів KNX та взаємодіє з мережею. При створенні функції потрібно описати “topic” та вказати, що функцію працює на запис даних до мережі. Основною частиною повідомлення до мережі є “payload”, який складається:

- “dstgad” – унікальний код компонента мережі (лампочка, температурний датчик);
- “dpt” – тип значення, який система передає до KNX мережі (потрібний код типу даних можна знайти на офіційному сайті KNX);
- “value” – значення, яке передається до мережі.

Система Node-RED надає можливість описувати та конфігурувати власний “payload” з використанням стандартних вузлів. Також система надає можливість створювати свої функції для динамічного створення повідомлення до KNX-мережі.

Створений вузол є обов'язковим для передачі повідомлення до мережі. Вузол може дублюватися та створюватися на кожен компонент мережі. Стан вузла та повідомлення при кожному запиті логуються та записуються в консоль веб-системи Node-RED. Усі повідомлення про помилки та непередбачувану поведінку вузла, веб-

система також логує в консоль або у файл (рисунок 4.2).



Рисунок 4.2 – Загальний вигляд нового вузла “knx-out”

Вузол “knx-in” був створений за принципом слухача. Система використовує вузол для отримання та обробки повідомлення від мережі. Усі повідомлення від мережі зберігаються у логах або файлах. відповідь від мережі має інший формат, ніж повідомлення до мережі (рисунок 4.3).

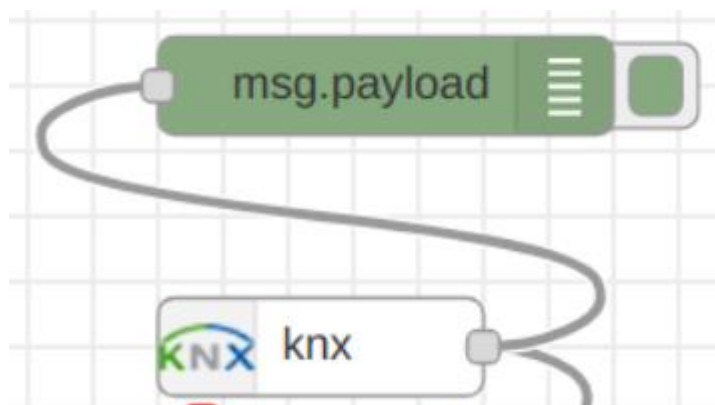


Рисунок 4.3 - Загальний вигляд нового вузла “knx-in”

Відповідь від мережі складається з (рисунок 4.4):

- “destination” – унікальний код компонента мережі (лампочка, температурний датчик);
- “rawValue” – стан компонента мережі.

```
{
  "payload": 0;
  "destination": "0/1/0",
  "rawValue": [0]
}
```

Рисунок 4.4 – Загальний вигляд відповіді від KNX мережі

Відповідь від мережі KNX приходить у шістнадцятковому форматі. Засобами

JavaScript та Node.js вузол конвертує вхідні дані у десятковий формат для коректної роботи з даними.

## 4.2. Веб-інтерфейс кінцевого користувача

Клієнт взаємодіє з сервером через підключення до Інтернету, а це означає, що підключення до Інтернету є головною умовою використання програмного забезпечення. Принцип REST використовувався для спілкування між клієнтом та сервером.

Коли клієнт переходить на будь-яку веб-сторінку, браузер надсилає GET-запит на сервер, сервер отримує запит, а після обробки надсилає відповідь, браузер отримує відповідь від сервера і відображає інформацію, отриману клієнтом

Веб-інтерфейс взаємодії з системою Node-RED, яка є адаптером між додатком клієнта та мережею KNX (рисунок 4.5).

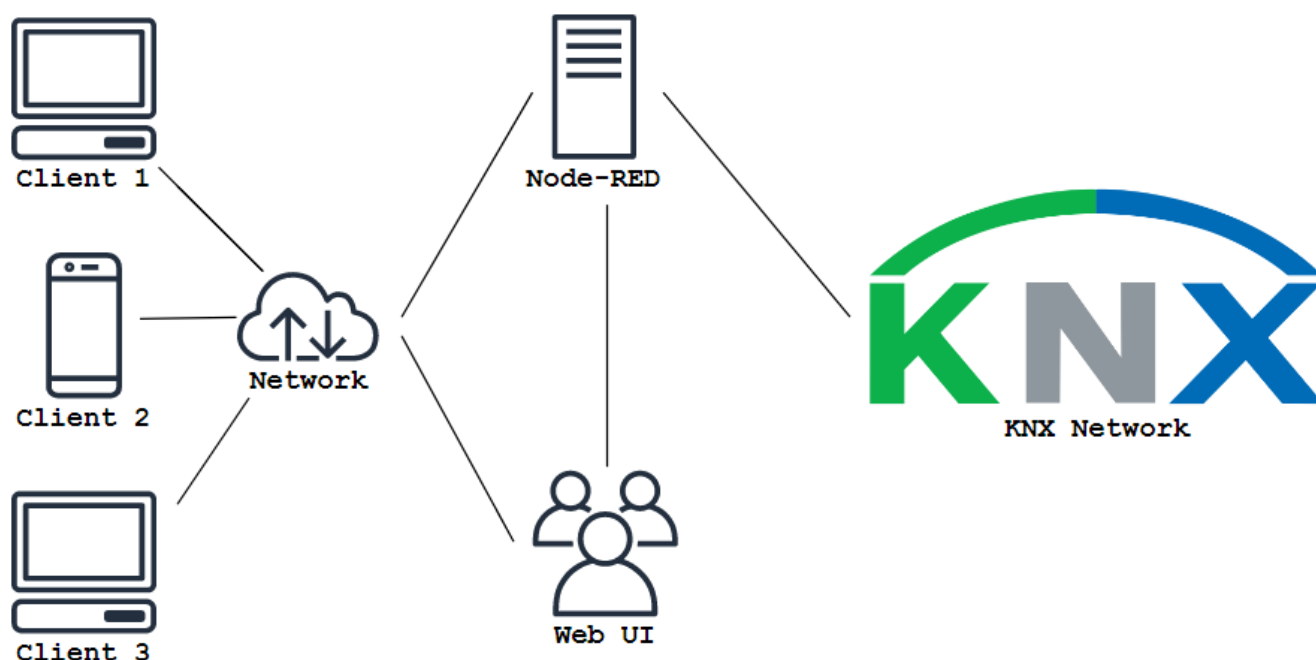


Рисунок 4.5 – Загальний вигляд системи

Для реалізації AJAX запитів було використано функції в JavaScript. Функції - ключова концепція в JavaScript. Найважливішою особливістю мови є першокласна

підтримка функцій (functions as first-class citizen). Будь-яка функція це об'єкт, і отже нею можна маніпулювати як об'єктом, зокрема:

- передавати як аргумент і повертати в якості результату при виклику інших функцій (функцій вищого порядку);
- створювати анонімно і привласнювати як значення змінних або властивостей об'єктів.

## **Висновки до розділу 4**

Для взаємодії користувача з мережею KNX було використано веб-систему Node-RED. Для підключення Node-RED та KNX мережі був створений адаптер та додано три додаткові вузли до системи Node-RED. Створені вузли надають можливість швидкої конфігурації алгоритму взаємодії KNX мережі з системою Node-RED. Під час розробки вузлів була додана можливість логування повідомлень до та з мережі. Нові вузли системи є обов'язковими елементами для створення особистого алгоритму взаємодії компонентів мережі. Клієнтська частина була розроблена за допомогою мови програмування JavaScript, бібліотеки jQuery та стилів CSS. Система кінцевого користувача взаємодії з мережею KNX через веб-систему Node-RED.



## 5. МЕТОДИ ВЗАЄМОДІЇ З ВЕБ-СИСТЕМОЮ

У цьому розділі описані системні вимоги до персонального комп'ютера для забезпечення стабільної та належної роботи розробленої системи, посібник користувача щодо її встановлення, а також детальний метод роботи користувача з програмним забезпеченням.

### 5.1. Системні вимоги

Щоб програмне забезпечення працювало належним чином, персональний комп'ютер повинен відповідати наступним мінімальним системним та апаратним вимогам:

- операційна система Linux ;
- будь-який браузер;
- процесор із частотою від 1.5 ГГц 32- або 64-розрядний;
- оперативна пам'ять від 2 ГБ

### 5.2. Інсталяція системи

Вимоги до встановленого програмного забезпечення:

- Node-RED;
- JavaScript;
- Node.js ;

Після встановлення “npm” та “Node.js”, адміністратору потрібно встановити веб-систему Node-RED, виконавши команду “sudo npm install -g --unsafe-perm node-red”.

Встановивши веб-систему для графічного програмування Node-RED, адміністратор повинен перейти в папку з додатковими розширеннями для Node-RED.

За замовчуванням в системі Linux ця папка має шлях “cd ~/.node-red/node\_modules”.

Далі потрібно встановити ряд додаткових модулів для коректної роботи адаптера у системі Node-RED виконавши команди:

- “sudo npm i @node-red/util”;
- “sudo npm i knx”;
- “sudo npm install --no-audit --no-update-notifier --save --save-prefix="~" --production [node-red-contrib-simplejs@0.0.4](#)”;
- “sudo npm install --no-audit --no-update-notifier --save --save-prefix="~" --production node-red-contrib-knx@0.8.2”;

Після встановлення додаткових модулів для Node-RED, система готова для встановлення адаптера між KNX мережею та Node-RED. Написаний модуль потрібно скопіювати у папку з додатковими модулями системи Node-RED. Після встановлення адаптеру, система Node-RED готова до запуску та використання її користувачами. Команда для запуску – “node-red”. Використовуючи інтерфейс Node-RED, потрібно завантажити у систему написаний алгоритм взаємодії “501\_flow.json”.

### 5.3. Сценарій роботи користувача з системою

Реалізована система складається з двох частин: веб-інтерфейс кінцевого користувача та веб-імплементация Node-RED системи. Веб-інтерфейс складається з двох частин: показання температурного датчика, який оновлюється при загрузці сторінки та контролера стану освітлення кімнати. При першій загрузці інтерфейсу, система зчитує стан температурного датчика та стан освітлення кімнати. (рисунок 5.1-2).



Рисунок 5.1 – Стану температурного датчика та стан освітлення, коли лампочка вклjučена.

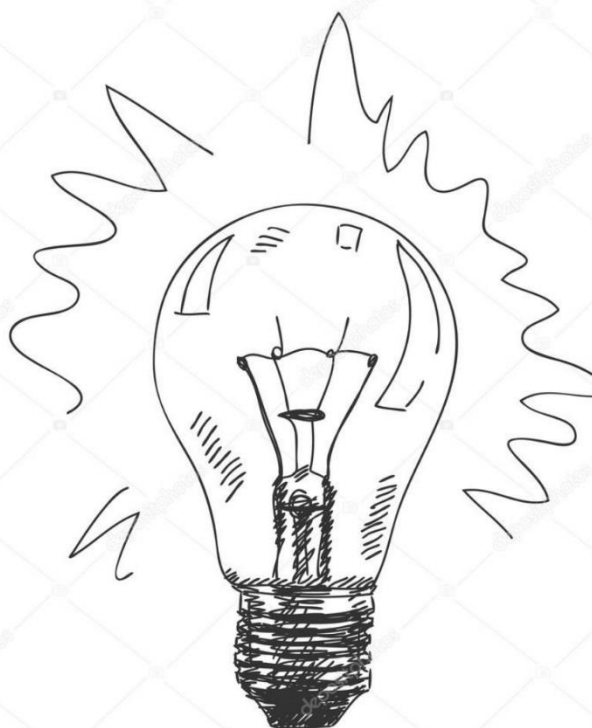


Рисунок 5.2 - Стану температурного датчика та стан освітлення, коли лампочка виключена.

Для зміни стану освітлення кімнати, потрібно натиснути на рисунок лампи. Стан температурного датчика оновлюється автоматично.

Друга частина системи складається з Node-RED системи та описаного алгоритму взаємодії компонентів KNX-мережі. Ця частина системи потрібна для технічного спеціаліста, який налаштовує взаємодію модулів, відслідковує стан компонентів системи, створює точки доступу до системи з веб-інтерфейсу кінцевого користувача та має доступ до KNX-мережі через веб-інтерфейс системи Node-RED. В імплементації адаптера між KNX-мережею та Node-RED системою було створено три додаткові вузлів для Node-RED: knx-controller, knx-in, knx-out. (рисунок 5.3).

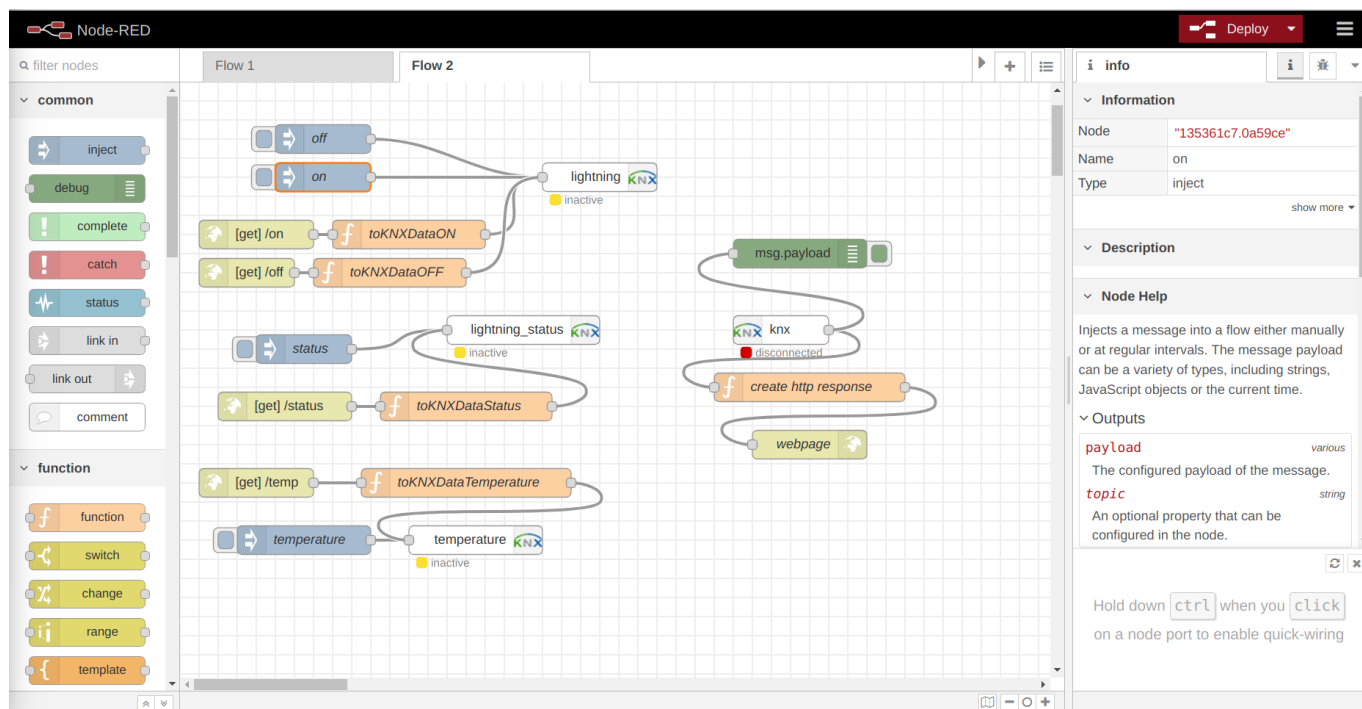


Рисунок 5.3 – Загальний вигляд створеного алгоритму взаємодії з новими вузлами системи

Головною частиною кожного алгоритму взаємодії компонентів мережі є knx-controller, який потрібно конфігурувати, вказавши порт підключення та IP-адресу KNX контролера. (рисунок 5.4)

Описаний продукт має три алгоритми взаємодії з KNX мережею. Кожен алгоритм має вихідний вузол, для формування відповіді від KNX мережі.

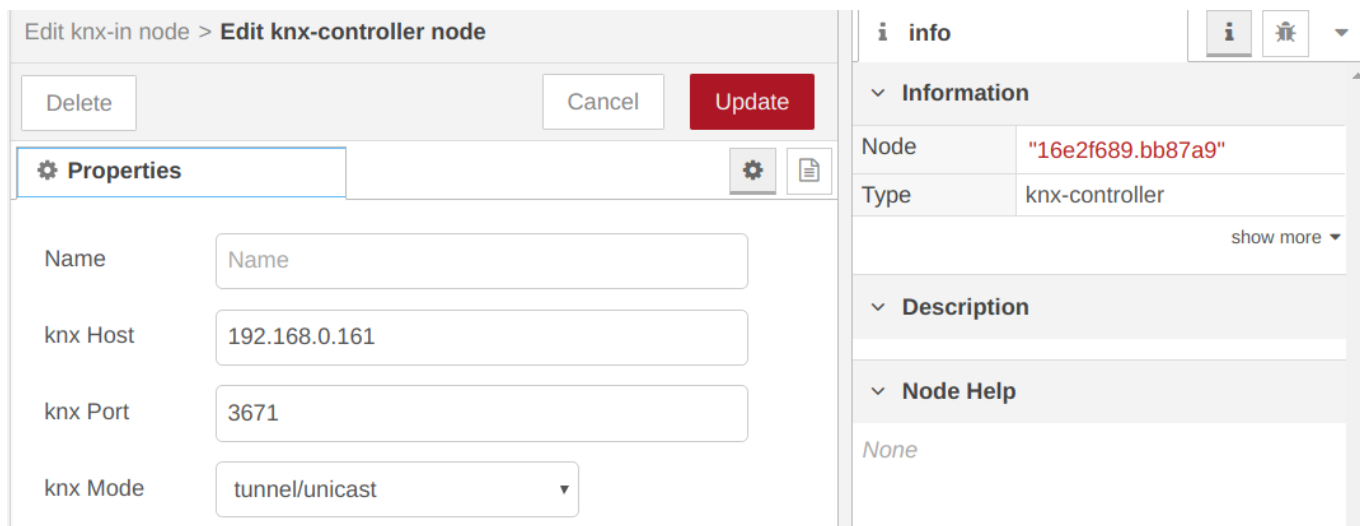


Рисунок 5.4 – Загальний вигляд налаштувань контролера Node-RED системи

Перший алгоритм має чотири вхідні вузла: вузол для включення освітлення через веб-інтерфейс Node-RED, вузол для включення освітлення через веб-інтерфейс кінцевого користувача через GET запит, вузол для виключення освітлення через веб-інтерфейс Node-RED, вузол для виключення освітлення через веб-інтерфейс кінцевого користувача через GET запит. (рисунок 5.5)

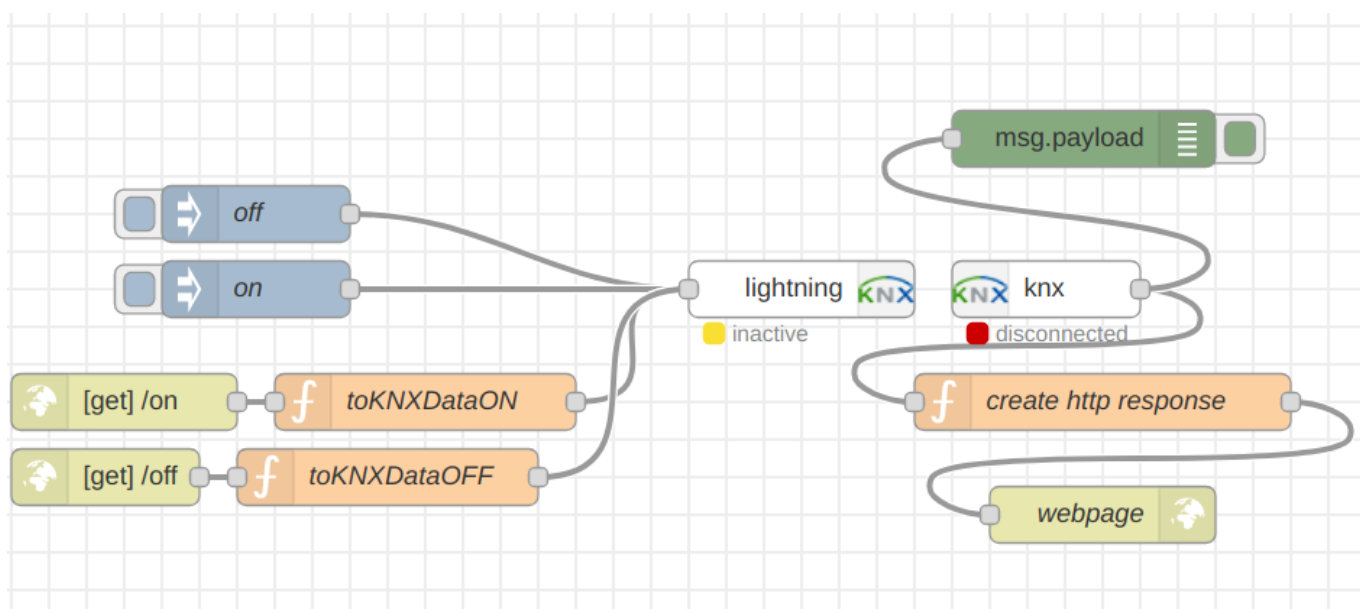


Рисунок 5.5 – Загальний вигляд алгоритму роботи зі зміною стану освітлення системи

Кожен вузол має JSON текст запиту до KNX мережі з кодом модуля та значення для зміни стану компонента мережі, найменування вузла системи та тип запиту. (рисунок 5.6)

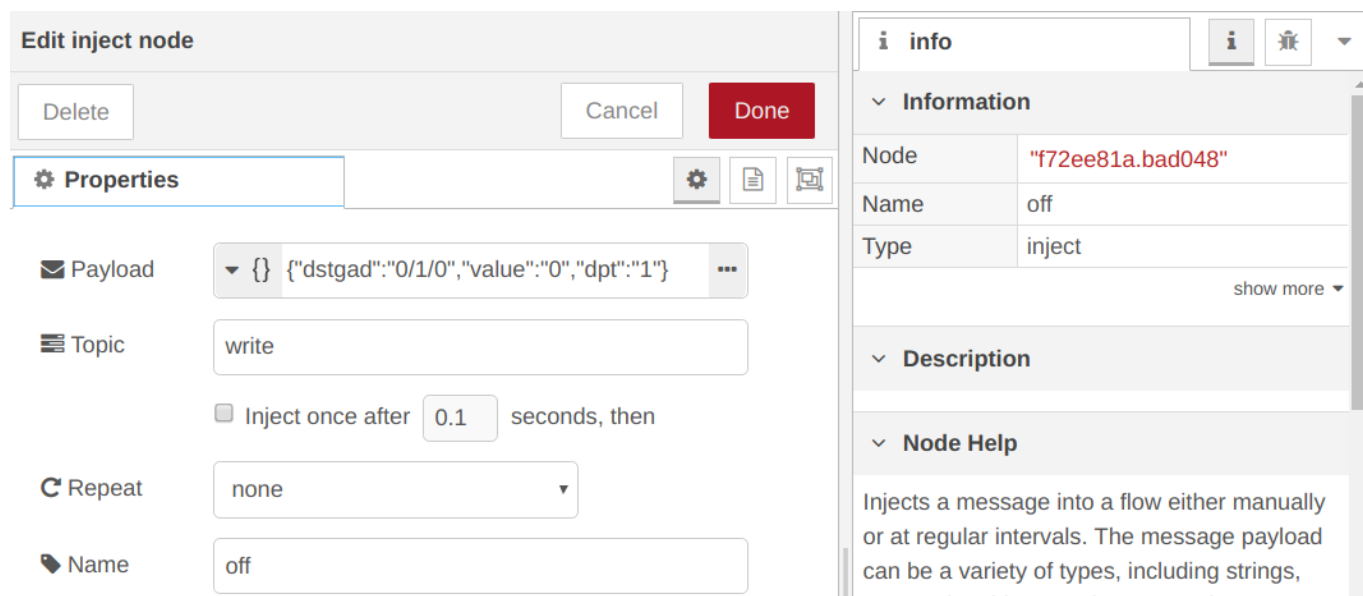


Рисунок 5.6 – Загальний вигляд конфігурації вузла для регулювання освітлення

Другий алгоритм має два вхідних вузла: вузол для зчитування стану освітлення через веб-інтерфейс Node-RED, вузол для зчитування стану освітлення через веб-інтерфейс кінцевого користувача через GET запит. (рисунок 5.7)

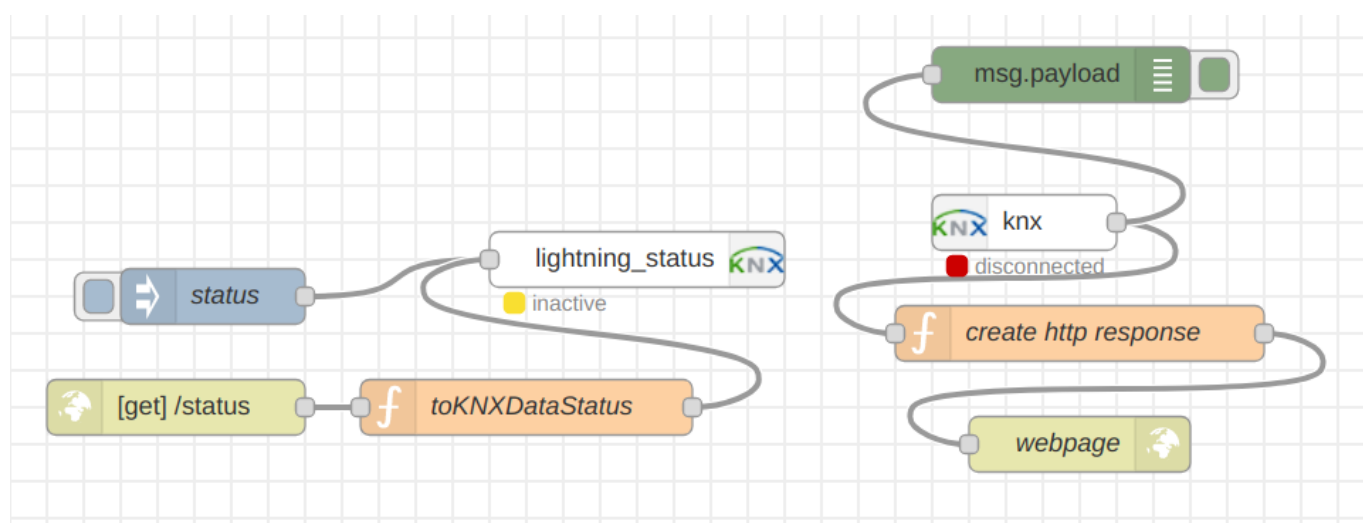


Рисунок 5.7 - Загальний вигляд алгоритму роботи зі зчитування стану освітлення системи

Вузол зчитування статусу освітлення не має значення для зміни стану системи, тому має тип запису читання. (рисунок 5.8)

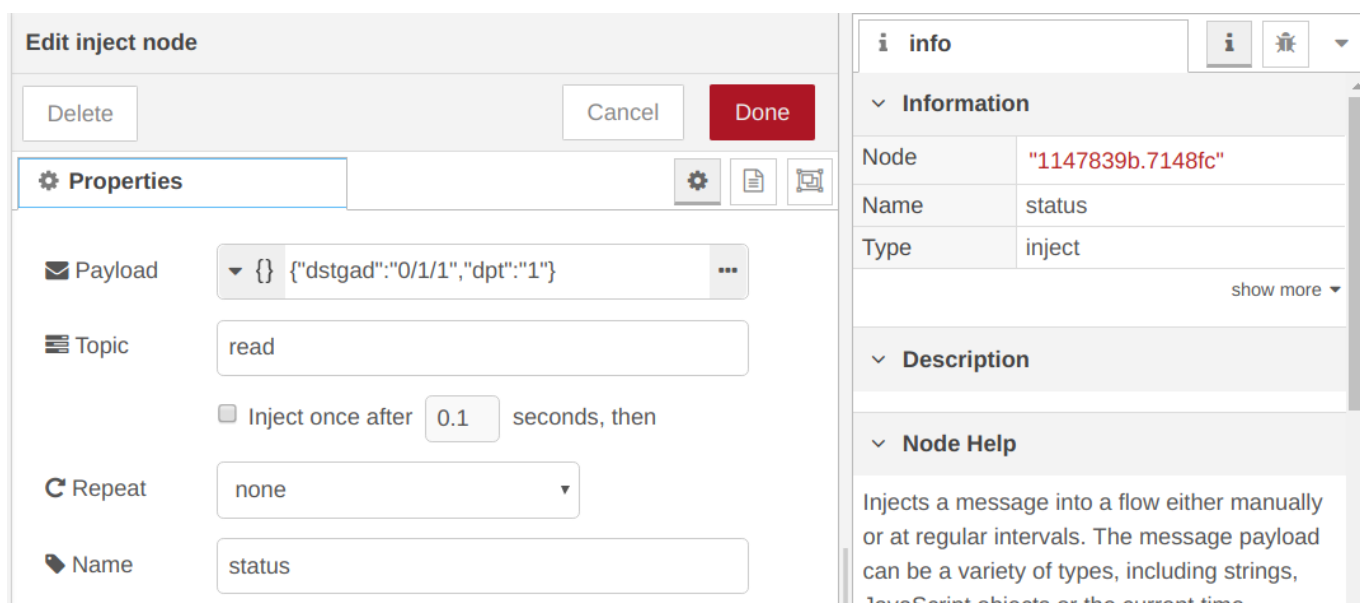


Рисунок 5.8 - Загальний вигляд конфігурації вузла для зчитування стану освітлення

Третій алгоритм має дві вхідні ноди: нода для зчитування стану температурного датчика через веб-інтерфейс Node-RED, нода для зчитування стану температурного датчика через веб-інтерфейс кінцевого користувача через GET запит (рисунок 5.9).

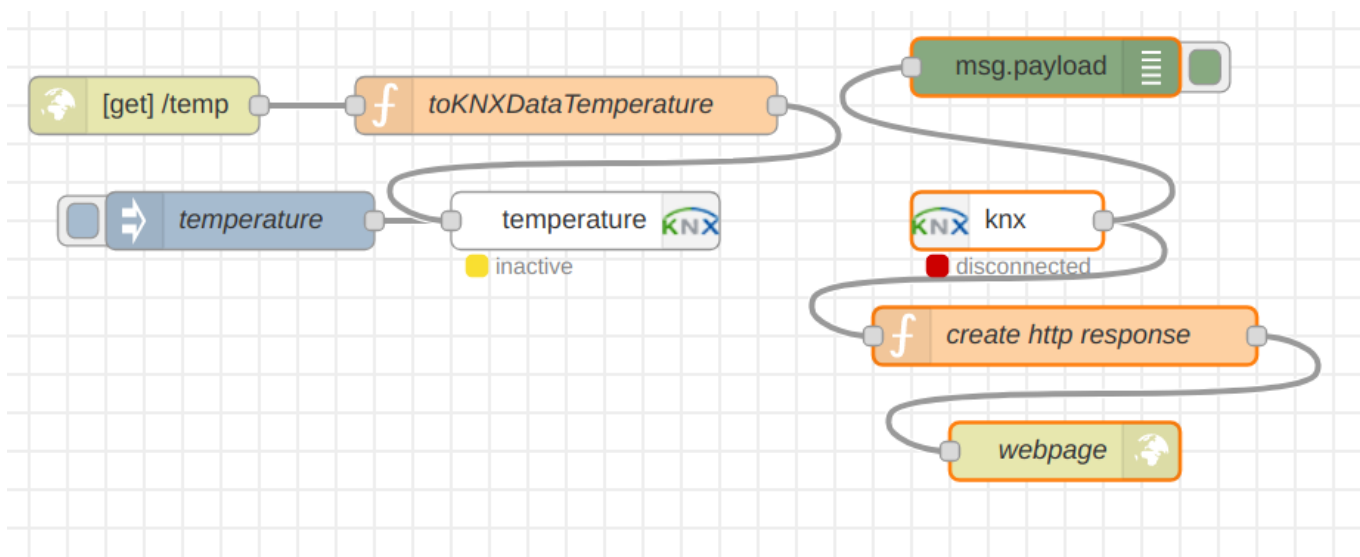


Рисунок 5.9 - Загальний вигляд алгоритму роботи зі зчитування стану температурного датчика системи

Вузол зчитування статусу температурного датчика не має значення для зміни стану системи, тому має тип запису читання. (Рисунок 5.10)

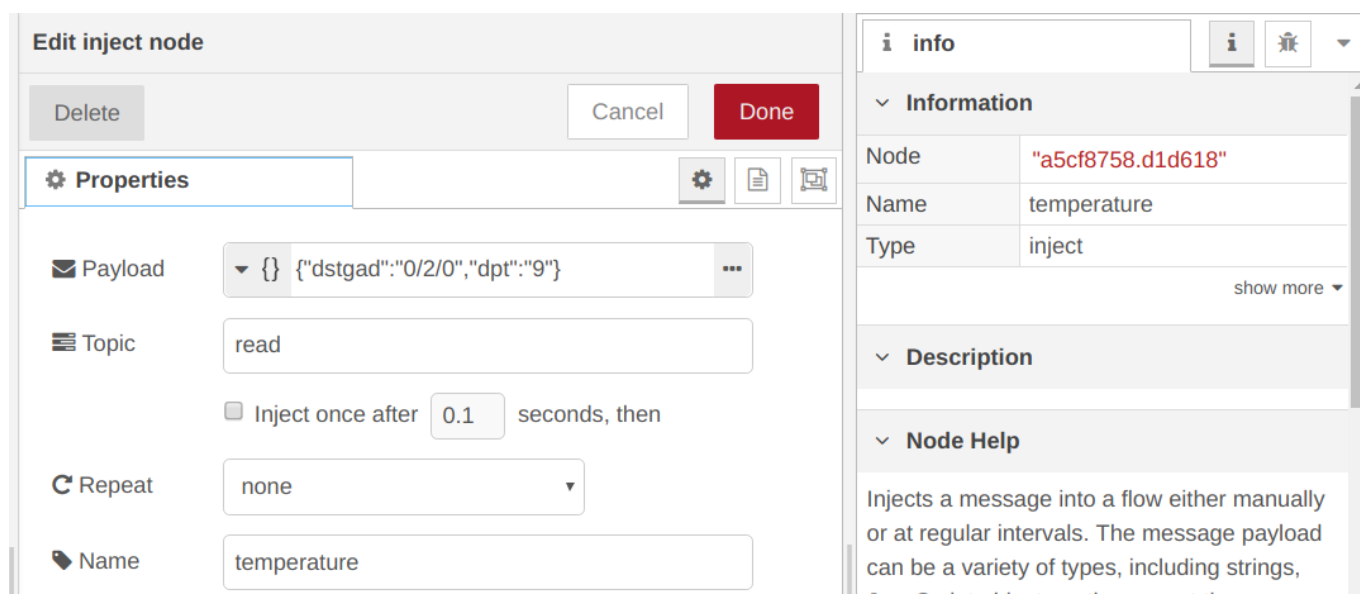


Рисунок 5.10 - Загальний вигляд конфігурації вузла для зчитування стану температурного датчика

## Висновки до розділу 5

Веб-інтерфейс реалізований у вигляді веб-застосунку, який є доступний з пристроїв з встановленим сучасним браузером. Інтерфейс користувача поділений на дві частини: веб-інтерфейс кінцевого користувача та інтерфейс технічного спеціаліста Node-RED. Застосунок має адаптивний дизайн, забезпечуючи належну роботу веб-програми на великих комп'ютерах та мобільних пристроїв.



## 6. РОЗРОБКА СТАРТАП ПРОЕКТУ

Ідея проекту - розробити реалізацію програмного забезпечення інтерфейсу людина-машина для візуалізації структури мережі KNX, контролю та моніторингу її компонентів. Ця система допоможе поєднати інтелектуальну програму управління кімнатою та програму моніторингу всіх змін у системі. Система дозволяє контролювати управління різними сумами без посилення на ваше місцезнаходження.

### 6.1. Опис ідеї стартап проекту

У підрозділі розглянуть наступні питання:

1. Ідея та її зміст;
2. Напрямки застосування;
3. Переваги для користувача.
4. Огляд існуючих аналогів.

Наступні пункти подаються у вигляді таблиці (таблиця 6.1) і відповідають на ряд поставлених питань.

Таблиця 6.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигода для користувача
Розробка програмної реалізації людино-машинного інтерфейсу для візуалізації KNX інсталяцій	Візуалізація структури KNX мережі	Можливість перегляду структури мережі та станів, у яких знаходяться її компоненти.
	Швидке налаштування алгоритму взаємодії компонентів мережі	Можливість налаштувати власний алгоритм взаємодії компонентів мережі, не маючи технічні знання

Для порівняння з конкуруючими пропозиціями товарів був проведений аналіз, який забезпечує:

1. Створення списку властивостей та функцій ідеї.
2. Пошук і аналіз конкурентних продуктів чи проектів, товарів-замінників чи товарів-аналогів, які вже існують у відкритому доступі на ринку .
3. Проводиться порівняльний аналіз показників, для встановлення ідеї визначаються показники що мають: а) гірші значення( W, слабкі), б) аналогічні(N, нейтральні) значення, в) кращі значення(S, сильні)

Створено список потенційних переваг по відношенню до продуктів-конкурентів.

Результат аналізу наведений у таблиці 6.2.

Таблиця 6.2. Визначення характеристик ідеї проекту

Техніко-економічні характеристики ідеї	Продукція конкурентів		Слабкі (W), нейтральні (N) та сильні (S) сторони		
	Gideon	Apple HomeKit	W	N	S
Назва продукту					
Операційна система та версії	Android, IOS	MacOS, IOS			+
Системні вимоги	Мінімальні	Мінімальні			+
Мови програмування	Java, Swift	Swift		+	
Необхідність встановлення додаткового ПЗ	наявність АПК	наявність АПК		+	
Ціна	\$600	\$500			+

В даний час аналогова система працює і представлена як мобільний додаток. Система підтримується на будь-якій платформі. Він не вимагає спеціальних системних вимог.

Аналогів створеної системи на вітчизняному ринку не знайдено.

## 6.2 Технологічний аудит ідеї проекту

Для технічного аудиту ідеї проекту було проведено аудит технологій. завдяки якому можна втілити ідею проекту. Аудит може визначити, чи є ці технології доступними, як їх можна вдосконалити [18]. Результат показаний у таблиці 6.3.

Таблиця 6.3. Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Розробка програмної реалізації людино-машинного інтерфейсу для візуалізації KNX інсталяцій	Середовище розробки JetBrains WebStorm	+	Доступна
	Node-RED	+	Доступна

Для реалізації було обрано середовище розробки WebStorm від компанії JetBrains, а також Node-RED веб-система для візуального програмування.

Вибрані технології не потребують вдосконалення та безкоштовні, і надають усі необхідні можливості для виконання цього завдання..

## 6.3 Аналіз ринкових можливостей запуску стартап проекту

Визначення ринкових можливостей, які можуть бути використані при ринковій реалізації проекту, та ринкові загрози, які можуть перешкоджати реалізації проекту, дозволяють планувати розробку проекту з урахуванням стану ринкового середовища, потреб потенційних замовників та пропозицій конкуруючих проектів.

Результати випробувань наведені в таблиці 6.4.

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

Показники стану ринку	Характеристика
Загальна потреба в продукції	Необхідна
Можливі річні обсяги випуску в натуральних показниках	Не обмежена
Річні обсяги випуску в вартісних показниках	1 млн \$
Динаміка ринку (якісна оцінка)	Зростання
Наявність обмежень для входу	Для роботи необхідний доступ до мережі Інтернет
Специфічні вимоги до стандартизації та сертифікації	Відсутні
Середня норма рентабельності в галузі (або по ринку)	60 відсотків

Підсумок аналізу потенційного ринку стартап-проекту є оптимістичною перспективою виходу на ринок найближчим часом, оскільки він все більше зацікавлений у ринку.

Внаслідок зростаючого попиту серед програмних продуктів слід запровадити більш зручну та автоматизовану систему створення програмного забезпечення та підвищення швидкості обчислень [19].

Потім визначаються потенційні групи клієнтів, їх характеристики та складається орієнтовний перелік вимог до продукції для кожної групи (таблиця 6.5).

Таблиця 6.5. Характеристика потенційних клієнтів стартап-проекту

Цільова аудиторія	Особливості поведінки споживачів	Вимоги споживачів до товару
-------------------	----------------------------------	-----------------------------

Власники “розумних” приміщень	Візуалізація структури KNX мережі та управління її компонентами	доступна ціна; зручність і простота використання; мобільність
Проектувальники KNX-мереж	Можливість швидкого налаштування алгоритму взаємодії компонентів мережі	зручність і простота у використанні

Далі слід аналіз, який визначає клієнтів або групу клієнтів, які є потенційними користувачами продукту. Необхідно провести аналіз питань, які допоможуть або перешкоджають реалізації проекту [20].

Результати представлені у таблицях 6.6 та 6.7 відповідно.

Таблиця 6.6. Фактори загроз

Фактор	Зміст загрози	Можлива реакція компанії
Поява конкурентів	Виникнення конкурентних проектів, які будуть фінансуватись компаніями-гігантами.	Оптимізація роботи, пришвидшення обчислень та додаткові можливості для користувача.
Зміни тенденцій ринку	Виникнення спеціалізованих програм на подібну тему	Адаптація програмного продукту, який буде містити в собі максимальну кількість рішень, які є актуальними на ринку.
Економічний спад	У клієнтів буде поступово падати попит	Акції, скидки на певний термін.
Зниження репутації компанії	Ситуація при якій клієнти перейдуть до більш успішної компанії.	Рекламні компанії, покращення продукту, виправлення помилок, що призвели до псування іміджу, формування механізмів захисту від подальшого псування іміджу.

Таблиця 6.7. Фактори можливостей

Фактор	Зміст загрози	Можлива реакція компанії
Невелика кількість конкурентів	Подібних програмних продуктів в нашій країні майже нема.	Піар свого проекту, реклама, презентації.
Зростання тенденцій ринку	Наразі зважаючи на ситуацію у країні попитом користуються технології, пов'язані з воєнною тематикою.	Швидка реклама та фокус на продажах, проведення зустрічей з клієнтами з презентаціями переваг проекту.
Підвищення іміджу компанії	Продукт є бажаним серед клієнтів	Активна реклама бренду

Наступний крок - визначити, яку конкуренцію очікувати та рівень конкуренції у товарах. Результати наведені в таблиці 6.8.

Таблиця 6.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії)
Тип конкуренції	Чиста Залежить від кількості конкурентів та якості	Підвищення якості продукту та зниження цін для встановлення балансу ціна-якість.
За рівнем конкурентної боротьби	Глобальна	Урахування культурних та регіональних чинників
За рівнем конкурентної боротьби	Локальна Конкуренція на вітчизняному ринку	Якщо конкурентних проектів на ринку немає, значить можна гратись з ціною на продукт.
Конкуренція за видами товарів	Створений товар може мати конкурентів, які	Розширення функціональних можливостей, реклама для

	пропонують аналогічний товар	популяризації програмного продукту
За характером конкурентних переваг	Цінова	Можливе підвищення ціни на додатковий функціонал.
За інтенсивністю	Марочна	Реклама позитивних сторін в продукті, у випадку виходу на зовнішній ринок

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (таблиця 6.9) на основі п'ятипорової моделі М. Портера, яка виділяє п'ять основних факторів, що впливають на привабливість вибору ринку через характер конкуренції:

- конкурент, що вже є у галузі;
- потенційні конкуренти;
- наявність товарів-замінників;
- постачальники, що конкурують за ринкову владу;
- споживачі, які конкурують за ринкову владу.

Таблиця 6.9. Аналіз конкуренції в галузі за М.Портером

Складові галузі	Прямі конкуренти в галузі	Потенційні конкуренти	Клієнти	Товари-замінники
	Продукти аналоги	Кращі продукти, ширший функціонал	Налагодження комунікації з клієнтами	Немає
Висновки	Інтенсивність конкурентної боротьби з боку прямих	Наявні усі можливості входу на ринок.	Необхідність клієнтської бази, тому важливо знаходити можливості	Без обмежень

	конкурентів незначна	Конкурентів немає	приваблення споживачів до власного продукту	
--	-------------------------	----------------------	---	--

На основі аналізу конкуренції, проведеного в таблиці 6.9 та з урахуванням особливостей дизайнерської ідеї (таблиця 6.2), вимог споживачів до товару (таблиця 6.5) та факторів маркетингового середовища (таблиці 6.6 та 6.7), на основі яких складається перелік конкурентних факторів (таблиця 6.10).

Таблиця 6.10. Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування
Невелика кількість конкурентів на ринку	На початок розробки проекту не виявлено конкурентів на ринку в нашій країні.
Доступність створеного продукту (програмно)	Для застосування слід бути підключеним до мережі.
Легкість і простота використання	Зрозуміліший інтерфейс. Документація для використання.
Підключення до мережі Інтернет	Слід підключитись до мережі, що б викачати базу.
Потреба у постійному супроводі	Відсутня
Додаткові компоненти	Не потрібні

Визначивши конкурентоспроможні фактори, можна визначити сильні та слабкі сторони товару, навпаки. Вони показані в таблиці 6.11.

Таблиця 6.11. Порівняльний аналіз сильних та слабких сторін проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
		-3	-2	-1	0	1	2	3
Мала кількість / відсутність конкурентів	15				+			



Системні вимоги	20			+				
Простота використання	15				+			
Не потрібен супровід	5	+						

Для SWOT-аналізу потрібна відповідна інформаційна підтримка, яка повинна включати: базу даних; методи та моделі, необхідні для SWOT-аналізу [22]. Методологія SWOT аналізу заснована на підході, який дозволяє досліджувати зовнішнє та внутрішнє середовище підприємства разом. Використовуючи цю техніку, ви можете встановити зв'язок між силою і слабкістю.

Таблиця 6.12. SWOT-аналіз проекту

Сильні сторони (S): розширений функціонал; ініціативна розробка; зручне масштабування;	Слабкі сторони (W): мале фінансування; затрати на датчики;
Можливості (O): вихід на міжнародні ринки; вдосконалення функціоналу; можливість імпорту нових мереж;	Загрози (T): незрозуміла тенденція попиту; поява конкурентів;

На основі аналізу SWOT розробимо альтернативну ринкову поведінку для запуску стартового проекту на ринку та приблизний оптимальний час їх реалізації на ринку з огляду на проекти потенційних конкурентів, які можуть бути запуснені на ринок [23]. Виявлені альтернативи аналізуються за часом та ймовірністю отримання ресурсів (таблиця 6.13).

Таблиця 6.13. Альтернативи ринкового впровадження стартап-проекту

Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
---------------------------------	--------------------------------	-------------------

Міжнародний ринок	Пошук клієнтів	Близько року
Розширення функціоналу	Пошук інвесторів	Від пів року

Тому програмне забезпечення чи дизайн слід випускати лише після ретельного аналізу ринку, конкурентів та потенційних загроз, які можуть виникнути.

## 6.4 Розробка ринкової стратегії проекту

Розробка ринкової стратегії в якості першого кроку вимагає визначення стратегії досягнення ринку: опис цільових груп потенційних споживачів (таблиця 6.14).

Таблиця 6.14. Вибір цільових груп потенційних споживачів

Опис цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в сегменті	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Власники “розумних” приміщень	Наявні	Існує попит	Замала	Незначна
Проектувальники KNX-мереж	Наявні	Існує попит, проте менший у порівнянні з власника	Низька	Помірна

Проаналізувавши та визначивши потенційних споживачів, автори ідеї проекту починають вибирати цільові групи [24]. Продукт буде запропонований саме їм. На основі аналізу буде визначена та обрана бажана стратегія [25].

Для роботи в обраних сегментах ринку необхідно сформулювати базову стратегію розвитку, яка визначена в таблиці 6.15. За результатами аналізу потенційних

споживчих груп (сегментів) автори ідей обирають цільові групи, для яких вони запропонують свій товар, та визначають стратегію виходу на ринок.

Таблиця 6.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Забезпечення ключових потреб клієнтів	Стратегія зосередження	Охоплення усіх ключових потреб клієнтів	Стратегія зосередження
Орієнтація поточної моделі на ринок приватних підприємств	Стратегія концентрованого маркетингу	Надання товару кращих властивостей та розширення функціоналу	Стратегія спеціалізації (спирається на диференціацію)

Наступним кроком є вибір стратегії, яка подана у таблиці 6.16.

Таблиця 6.16. Визначення базової конкурентної поведінки

Чи є проект новий на ринку	Так
Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Так
Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Ні
Стратегія конкурентної поведінки	Стратегія заняття конкурентної ніші

Стратегія конкурентної ніші була реалізована, оскільки її найчастіше використовують компанії, які займають друге чи третє місце серед лідерів ринку.

Виходячи з вимог споживачів від обраних сегментів до постачальника та програмного забезпечення, слід розробити стратегію позиціонування (таблиця 6.17), яка передбачає формування ринкової позиції, за якою споживачі повинні визначити бренд чи дизайн [23].

Таблиця 6.17. Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту
Доступна ціна, простота і зручність використання	Стратегія диференціації	Доступність через ціну та технічні характеристики. Можливість гнучкого налаштування через веб-інтерфейс.	стандарти якості; метрики програмного забезпечення

## 6.5 Розробка маркетингової програми стартап проекту

Розглянемо процес розробки маркетингової програми для нашого стартап-проекту. Концепції маркетингу товарів повинні бути визначені та сформовані. У таблиці 6.18 узагальнено проведені дослідження та їх результати щодо конкурентоспроможності продукції. [22]

Таблиця 6.18. Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
Оцінка якості ПП	Спрощення бізнес процесів. Надання можливості планування доходів	Розрахункові показники, точність та достовірність яких можна оцінювати; самостійність програмної системи.

Далі нам потрібно розробити трирівневу модель маркетингу товару: пояснити ідею товару, його фізичні елементи та особливості процесу його доставки (таблиця 6.19).

Таблиця 6.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
Товар за задумом	Розробка програмної реалізації людино-машинного інтерфейсу для візуалізації KNX інсталяцій
Товар у реальному виконанні	Властивості/характеристики
	Реалізовано візуалізацію структури KNX мережі з використанням веб-системи для графічного програмування Node-RED, з можливостями створення власного алгоритму взаємодії компонентів мережі
Товар із підкріпленням	До продажу: стандартна розроблена
	Після продажу: додані додаткові можливості

Наступним кроком є встановлення цінових лімітів, яких слід дотримуватися при визначенні ціни потенційного товару (таблиця 6.20).

Таблиця 6.20. Визначення меж встановлення ціни

Рівень цін на товарианалоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни
50 – 1000 \$	1000 – 5000 \$	10 – 15 \$

Наступним кроком є визначення оптимальної системи розподілу, в якій приймається рішення (таблиця 6.21)

Таблиця 6.21. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Система повинна надаватися в режимах пробної версії та повної сплатити після закінчення випробувального строку	Проста та легкість в експлуатації	Веб-сайт	Вертикальна маркетингова система

Останнім елементом маркетингової програми є розробка концепції маркетингової комунікації, яка базується на заздалегідь обраній основі позиціонування, конкретної поведінки клієнта (таблиця 6.22).

Таблиця 6.22. Концепція маркетингових комунікацій

Поведінка цільових клієнтів	Канали комунікацій цільових клієнтів	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення
Бажання отримати більше за менші гроші	Веб-сайт	Низька ціна Легкий і простий у використанні продукт	Довести, що програмний продукт оптимально працює з різними KNX-мережами та надає можливість створювати власні алгоритми взаємодії компонентів мережі

## **Висновки до розділу 6**

Цей розділ формулює ідею стартап-проекту, проводить технологічний аудит, аналізує ринкові можливості запуску стартап-проекту, розробляє ринкову стратегію та маркетингову програму. Перевага проекту полягає в тому, що в нашій країні на ринку немає конкурентів, і тому конкуренція не виявлена. На даний момент попит на товар високий, адже мережі KNX набирають популярності в галузі автоматизації будівництва та будівництва.

Перспективи впровадження з огляду на потенційні групи користувачів, стан конкуренції та конкурентоспроможності проекту – прямі, і тільки доводять можливість впровадження, та не марну розробку створеного програмного продукту.

## ВИСНОВКИ

Метою магістерської дисертації є розробка програмної реалізації людино-машинного інтерфейсу для візуалізації KNX інсталяцій. Відповідно було проаналізовано основні принципи проектування інтерфейсу користувача та проблематику проектування інтерфейсу користувача. За основу візуалізації KNX інсталяцій було обрано веб-систему для графічного програмування Node-RED, яка надає можливість описати взаємодію компонентів мережі у вигляді блок схеми, не потребуючи детальних технічних знань у сфері програмування та мереж. Ознайомлено зі стандартами KNX-мереж, з можливостями та способами налаштування роботи “розумної” мережі.

Враховуючи вимоги до зручності підтримки, налаштування, зручності, безпеки та гнучкості сучасних систем автоматизації компонентів будівлі, було розроблено людино-машинний інтерфейс для візуалізації KNX-мережі, контролю її компонентів та швидкого налаштування алгоритму взаємодії компонентів мережі.

Розроблена система забезпечує наступну функціональність:

- встановлення зв'язку між Node-RED та KNX мережею;
- відображення елементів управління мережі;
- управління та налаштування алгоритму взаємодії компонентів KNX мережі через інтерфейс Node-RED;
- управління станом модулів мережі через веб-інтерфейс кінцевого користувача;
- підключення до мережі;
- відображення історії користування;
- відображення подій мережі у реальному часі.

Головним компонентом системи є адаптер між веб-системою Node-RED та KNX мережею. Він додає три додаткові вузли до вже існуючих системи графічного програмування та надає можливість коректного налаштування алгоритму взаємодії



компонентів мережі. Node-RED надає можливість налаштовувати мережу з будь-якого пристрою.

Веб-інтерфейс кінцевого користувача реалізований у вигляді веб-застосунку, що може бути доступний з будь-якого браузера. Застосунок має адаптивний дизайн, що дозволяє користуватися ним з будь-якого пристрою.

Для реалізації поставленої задачі було використано сучасні технології, котрі є одними з найпопулярніших у світі, що забезпечує зручну підтримку проекту та надає можливість легкого масштабування. Для розробки адаптера проекту було використано платформу Node.js та платформу Node-RED, які надають інструменти для конфігурації існуючих платформ. Для реалізації користувацького інтерфейсу були використані мова програмування JavaScript та бібліотека для підтримки AJAX запитів – jQuery. Для розмітки веб-сторінки використано HTML та стилі CSS.

Було описано роботу з програмним продуктом. У розділі є інструкція роботи з програмою. Зазначено всі можливі функції, які може використовувати користувач програмного продукту.

Даний програмний продукт має суттєву перевагу над існуючими аналогами та конкурентами, які представлені на внутрішньому ринку. Продукт має шанси розвиватись, наповнюватись новим функціоналом, а отже було визначено основні маркетингові стратегії. Потенційними користувачами програмного продукту є інженери-проектувальники KNX-мереж та користувачі, які не є спеціалістами у даній області

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технология KNX для систем автоматизации [электронный ресурс] – режим доступа: <https://www.ixbt.com/home/knx-intro.shtml>
2. Основні поняття ІК та засоби їх проектування [Електронний ресурс] - Режим доступу: <http://elar.khnu.km.ua/jspui/bitstream/123456789/1415/2/Rozdil1.pdf>
3. Тараненко К.Г. Автоматизований аналіз та оцінка зручності використання програмних систем. Системний аналіз та інформаційні Системний аналіз та інформаційні технології / Тараненко К.Г., Гученко І.В. : матеріали 12-ї Міжнародної Науково-технічної конференції SAIT 2010, Київ, 25–29 травня 2010 р. / ННК “ІПСА” НТУУ “КПІ”. – К.: ННК“ІПСА” НТУУ “КПІ”, 2010. – 544 с.
4. EST - Опис [Електронний ресурс] — Режим доступу: <http://www.konnexrussia.ru/knx-standard/knx-tools/ets/>
5. Технология KNX для систем автоматизации [Электронный ресурс] – режим доступа: <https://www.ixbt.com/home/knx-intro.shtml>
6. Стандарт KNX-мереж [Електронний ресурс] — Режим доступу: <http://www.gudramaja.com/content/ru/296/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-%D0%B2-%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82-KNX.html>.
7. Node-RED – Режим доступу: <https://nodered.org/>
8. JetBrains WebStorm – Опис [Електронний ресурс] — Режим доступу: <https://itpro.ua/product/jetbrains-webstorm/?tab=description>.
9. Современный учебник JavaScript – Опис [Електронний ресурс] — Режим доступа: <https://idg.net.ua/blog/uchebnik-css>.
10. Node.js для начинающих – Опис [Електронний ресурс] — Режим доступа: <http://www.nodebeginner.ru/>
11. Tutorialspoint Node.js – Опис [Електронний ресурс] — Режим доступа: [https://www.tutorialspoint.com/nodejs/nodejs\\_tutorial.pdf](https://www.tutorialspoint.com/nodejs/nodejs_tutorial.pdf).

12. Node-RED – Опис [Электронный ресурс] — Режим доступа: <https://r-iot.org/tag/node-red/>
13. jQuery учебник – Опис [Электронный ресурс] — Режим доступа: <http://www.wisdomweb.ru/JQ/jquery-first.php>
14. The global structure of an HTML document – Опис [Электронный ресурс] — Режим доступа: <https://www.w3.org/TR/html401/struct/global.html>.
15. Современный учебник по CSS document – Опис [Электронный ресурс] — Режим доступа: <https://learn.javascript.ru/>
16. Основы document – Опис [Электронный ресурс] — Режим доступа: <https://sasscss.ru/guide/>
17. The global structure of an HTML document – Опис [Электронный ресурс] — Режим доступа: <https://www.w3.org/TR/html401/struct/global.html>.
18. Черкасов Д. О., Сайбель Н. Ю. Стартап: характеристика понятия и этапы развития // Современное состояние и перспективы развития научной мысли: сборник статей Международной научно-практической конференции (25 мая 2015 г., г. Уфа) в 2 ч. – Уфа: АЭТЕРНА, 2015. – Ч. 1. – С.141-143.
19. Розроблення стартап-проекту: Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О. А. Гавриша. – Київ: НТУУ «КПІ», 2016. – 28 с
20. Харниш, В. Правила прибыльных стартапов : как расти и зарабатывать деньги / В. Харниш ; пер. с англ. В. Хозинского. – Москва : Манн, Иванов и Фербер, 2012. – 279 с.
21. Тиль, П. От нуля к единице : как создать стартап, который изменит будущее / П. Тиль, Б. Мастерс; перевод с англ. – Москва : Альпина паблишер, 2015. – 188 с. Петруненко А. Оценка коммерческой привлекательности проекта [Электронный ресурс].— Режим доступа: <http://www.techbusiness.ru/tb/archiv/number2/page01.html>
22. Каталог автоматизированных систем [Электронный ресурс]. – Электрон. дан. - Режим доступа: <http://www.erponline.ru/software/open/>

23. Филип Котлер, Роланд Бергер, Нильс Бикхофф. Стратегический менеджмент по Котлеру. Лучшие приемы и методы = The Quintessence of Strategic Management: What You Really Need to Know to Survive in Business. — М.: Альпина Паблишер, 2012. — 144 с

24. Бариленко В. И. Бизнес-анализ как важный вид консалтинговых услуг // РИСК: Ресурсы, Информация, Снабжение, Конкуренция. — № 4. — 2012. — С.202-207.

25. Квашнин А. Как продвигать проекты коммерциализации технологий: серия методических материалов «Практические руководства для центров коммерциализации технологий» / М. Катешова, А. Квашнин, под рук. П. Линдхольма, проект EuropeAid «Наука и коммерциализация технологий», 2006. — 52 с.

## ДОДАТОК А

### Апробації

Розробка програмної реалізації людино-машинного інтерфейсу  
для візуалізації KNX інсталяцій

УКР.НТУУ"КПІ" \_ТЕФ\_АПЕПС\_ ТМ4120\_19М

Аркушів 3

2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

# СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVII Міжнародної  
науково-практичної конференції  
молодих вчених та студентів  
м. Київ, 23-26 квітня 2019 року,

ТОМ 2



Київ- 2019

## ВІЗУАЛІЗАЦІЇ СТРУКТУРИ KNX-МЕРЕЖІ З ВИКОРИСТАННЯМ ЛЮДИННО-МАШИННОГО ІНТЕРФЕЙСУ

В сучасному світі кожна людина очікує більше зручності та комфорту, як у себе вдома так і на роботі. Сьогодні KNX є одним з найбільш поширених рішень для використання в середніх і великих системах автоматизації будинків, офісів і комерційних приміщень. Вимоги до зручності, безпеки та гнучкості сучасних будівель постійно зростають [1]. При цьому, враховуючи постійний технологічний розвиток та підвищення автоматизації систем забезпечення комфорту, найбільш важливим питанням стає економія електроенергії. Саме тому попит на інтелектуальні системи, що відповідають цим умовам, постійно зростає. Технологія KNX знайшла широке застосування в області управління квартирами і будівлями по всьому світу. У світі існує більше 100 компаній-членів Асоціації KNX, які пропонують у своїх каталогах майже 7000 груп KNX- сертифікованих продуктів для різних застосувань.

Використання KNX забезпечує реальні переваги для архітекторів, проектувальників і виробників робіт, а також, в першу чергу, для власників і/або користувачів будівель, а саме:

**Економія часу.** Зв'язування усіх пристроїв, що обмінюються між собою інформацією, за допомогою однієї загальної шини помітно скорочує час проектування системи і час її установки. Унікальний інженерний програмний пакет ETS, загальний для усіх виробників і програмних застосувань, дозволяє здійснювати проектування, відладку і налаштування систем, що містять у своєму складі KNX- сертифіковані елементи. Оскільки цей програмний пакет є єдиним для усіх виробників, інтегратори систем можуть об'єднувати в проєкті продукти від різних виробників, що використовують різні канали для обміну інформацією (виті пари, радіочастотні канали, електричні лінії або IP/Ethernet), у складі єдиної системи.

**Гнучкість і здатність адаптуватися до майбутніх змін.** KNX- система може бути легко пристосована до виконання нових завдань і може бути легко розширена. Нові компоненти можна з легкістю підключати до вже працюючої системи.

**Низькі експлуатаційні витрати і значне зниження енергоспоживання.** Освітлення і опалювання включаються тільки тоді, коли вони дійсно потрібні, наприклад, відповідно до заданих тимчасових програм і/або лише при реальній присутності людей, що дозволяє економити і електроенергію, і фінансові кошти. Більше того, освітлення може контролюватися автоматично відповідно до реальної інтенсивності денного світла, що допомагає забезпечувати мінімальний необхідний рівень яскравості освітлення на кожному робочому місці і знижувати енергоспоживання (включеними залишаються тільки ті джерела освітлення, які дійсно потрібні).

Використання технології KNX може надати рішення, які при використанні звичайних методів створення подібних систем можуть бути реалізовані лише насилу. Контроль усіх застосувань в квартирі або будівлі може здійснюватися з однієї сенсорної панелі. Починаючи з систем опалювання, вентиляції і контролю доступу, і закінчуючи дистанційним керуванням усіма побутовими електроприладами - KNX відкриває абсолютно нові шляхи для підвищення комфорту, безпеки і економного енергоспоживання в квартирах і будівлях.

Перелік посилань:

1. Технологія KNX для систем автоматизації [електронний ресурс] – режим доступу: <https://www.ixbt.com/home/knx-intro.shtml>

<b>Використання методів виявлення автоматичних моделей поведінки для побудови аналітичних сценаріїв.</b>	122
<i>ОЛЕНЄВА К.М., аспірант;</i>	
<i>ШПУРИК В.В.</i>	
<i>Керівник - доц., к.т.н. Коваль О.В.</i>	
<b>Візуалізації структури KNX-мережі з використанням людинно-машинного інтерфейсу.</b>	123
<i>ЯШИН А.С., магістрант гр. ТМ-81мп</i>	
<i>Керівник - доц., к.е.н. Сегеда І.В.</i>	
<b>Розробка агента моніторингу і управління попитом на електричну енергію "Розумної будівлі".</b>	124
<i>ШАРНІН С.А., магістрант гр. ТІ-81мп</i>	
<i>Керівник - доц., к.т.н. Ковальчук А.М.</i>	
<b>Агент моніторингу та управління режимами роботи мікроенергостанцій.</b>	125
<i>СТОЛЯР А.В., магістрант гр. ТВ-81мп</i>	
<i>Керівник - доц., к.т.н. Ковальчук А.М.</i>	
<b>Розробка серверної частини для веб-додатку відкритий спортивний майданчик з е-сервісами.</b>	126
<i>СЕРБІН А.В., магістрант гр. ТВ-82</i>	
<i>Керівник - доц., к.т.н. Ковальчук А.М.</i>	
<b>Розв'язок задачі обрахунку водонагрівача в інтерактивному режимі з використанням клієнт-серверної архітектури.</b>	127
<i>РОМАНОВ О.В., магістрант гр. ТВ-61мп</i>	
<i>Керівник - доц., к.т.н. Кузьменко І.М.</i>	
<b>Оцінка територій для побудови вітроелектростанцій із застосуванням мультиагентних технологій та ГІС.</b>	128
<i>ПІДВИШЕННИЙ Т.О., магістрант гр. ТВ-81мп</i>	
<i>Керівник - асист. Швайко В.Г.</i>	
<b>REST-інтерфейс як основа комунікації систем контролю доступу.</b>	129
<i>ОБРУСНІК Д.В., студент гр. ТВ-82мп</i>	
<i>Керівник - доц., к.е.н. Левченко Л.О.</i>	
<b>Інструментальні засоби аналізу впливу параметрів експерименту на сигнатуру морського об'єкту.</b>	130
<i>ОБІЩЕНКО А.А., магістрант гр. ТВ-81мп</i>	
<i>Керівник - доц., к.т.н. Варава І.А.</i>	
<b>Система моделювання структури та функціонального контенту інженерних систем енергоефективної будівлі.</b>	131
<i>КУРСЕНКО Л.О., магістрант гр. ТІ-81мп</i>	
<i>Керівник - доц., к.т.н. Шпурик В.В.</i>	
<b>Інструментальний засіб підтримки динамічного реєстру інформаційних ресурсів на базі ОРБД Caché.</b>	132
<i>КОСТЕНКО О.П., магістрант гр. ТВ-81мп</i>	
<i>Керівник - доц., к.т.н. Кублій Л.І.</i>	
<b>Мультиагентні системи в децентралізованих мережах енергоспоживання.</b>	133
<i>ЖОРНОВИЙ Е.Г., магістрант гр. ТВ-82</i>	
<i>Керівник - доц., к.т.н. Ковальчук А.М.</i>	
<b>Розробка веб-додатку для відкритого спортивного майданчику з е-сервісами.</b>	134
<i>АМБРОС С.М., магістрант гр. ТВ-81мп</i>	
<i>Керівник - доц., к.т.н. Ковальчук А.М.</i>	